



Den digitale revolution

fortællinger fra datalogiens verden

Naur, Peter; Vinter, Brian; Hansen, Klaus; Mogensen, Torben Ægidius; Erleben, Kenny; Pisinger, David; Nielsen, Mads; Kringelbach, Morten; Pedersen, Esben Warming; Blume, Peter; Helles, Rasmus; Andersen, Tariq Osman; Bansler, Jørgen P.; Clausen, Hasse Rehding; Jensen, Inge Hviid; Zachariasen, Martin

Publication date:
2010

Document version
Også kaldet Forlagets PDF

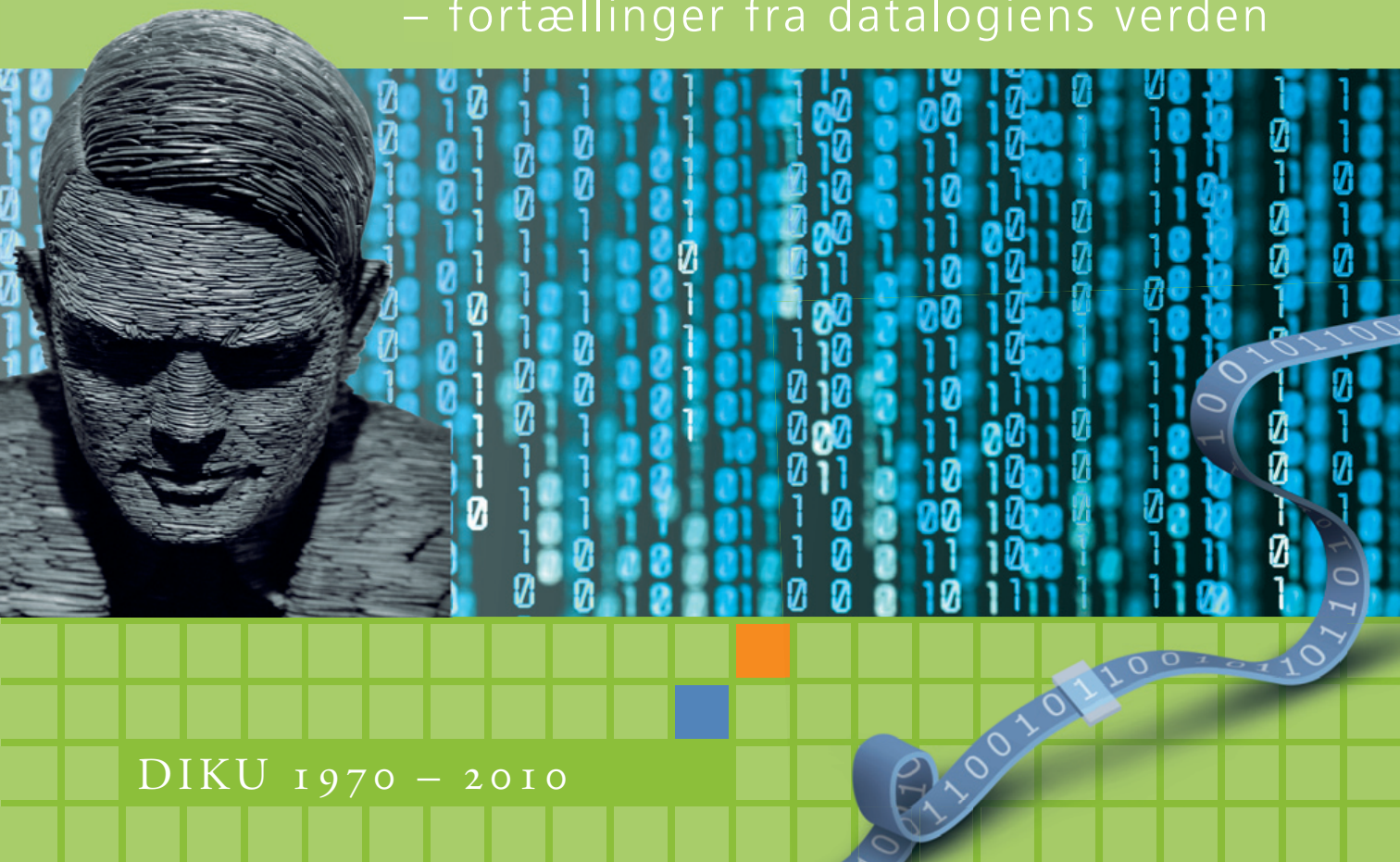
Document license:
[CC BY-NC-ND](#)

Citation for published version (APA):
Naur, P., Vinter, B., Hansen, K., Mogensen, T. Æ., Erleben, K., Pisinger, D., Nielsen, M., Kringelbach, M., Pedersen, E. W., Blume, P., Helles, R., Andersen, T. O. (red.), Bansler, J. P. (red.), Clausen, H. R. (red.), Jensen, I. H. (red.), & Zachariasen, M. (red.) (2010). *Den digitale revolution: fortællinger fra datalogiens verden*. Datalogisk Institut. http://www.diku.dk/ominstituttet/jubilee/dikus_jubilaemsskrift/0.DIKU_bog_72dpi.pdf



Den digitale revolution

– fortællinger fra datalogiens verden



DIKU 1970 – 2010

Abstract geometric lines in the top right corner and a blue ribbon with binary code (0s and 1s) on the right side.

Den digitale revolution – fortællinger fra datalogiens verden

Datalogisk Institut, Københavns Universitet (DIKU)

September 2010

Den digitale revolution – fortællinger fra datalogiens verden

Bogen er udgivet af Datalogisk Institut, Københavns Universitet (DIKU) i anledning af instituttets 40 års jubilæum med bidrag fra forskere tilknyttet instituttet.

Redaktion:

Tariq Andersen, phd-studerende, Jørgen Bansler, professor, Hasse Clausen, lektor, Inge Hviid Jensen, kommunikationsmedarbejder og Martin Zachariasen, institutleder.

Forsidemotiv: Foto af skulptur af Alan Turing, © basegreen lokaliseret på flickr.com/photos/basegreen

Oplag: 1000 eks.

Grafisk design og produktion: Westring + Welling A/S

ISBN: 978-87-981270-5-5

© Datalogisk Institut 2010. Citater er tilladt under creative commons.



INDHOLD

DEN DIGITALE VERDEN – ET BARN AF OPLYSNINGSTIDEN 6

Datalogi – læren om data // *Peter Naur* 10

DEN TEKNISKE PLATFORM 16

Supercomputeren – mange bække små... // *Brian Vinter* 22

Internetpionerer i Danmark // *Klaus Hansen* 34

Nogle programmeringssprog er mere lige end andre // *Torben Mogensen* 50

DEN DIGITALE TEKNIKS MULIGHEDER 64

Simulering og modellering af robotter og mennesker // *Kenny Erleben* 68

Optimering – et grundlag for bæredygtig it // *David Pisinger* 80

Machine learning – automatisk læring med computere // *Mads Nielsen* 92

Hjernen, den emotionelle computer? // *Morten Kringelbach* 104

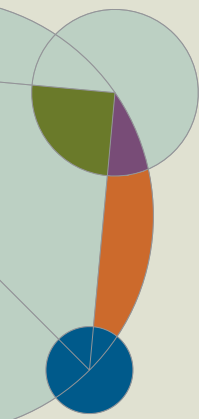
MENNESKERS OMGANG MED DEN DIGITALE TEKNIK 116

Tangible computing: musens død? // *Esben Warming Pedersen* 120

Krav til informationssikkerhed // *Peter Blume* 130

Digitale medier // *Rasmus Helles* 142

Sponsorer 154



Forord

Dette skrift udgives i anledning af, at Datalogisk Institut, Københavns Universitet (DIKU), fylder 40 år. Da DIKU blev oprettet i 1970 – som det første institut for datalogi i Danmark – kunne man dårligt nok ane konturerne af det, der i dag kan kaldes den digitale revolution. I dag, 40 år senere, er den udbredte anvendelse af informationsteknologi i gang med at ændre den måde, vi indretter vores samfund på, samtidig med at den giver mennesker over hele kloden helt nye muligheder for at kommunikere og samarbejde.

DIKUs 40 års jubilæumsskrift er ikke kun et historisk tilbageblik til dengang, da datamaskinerne var få, store, dyre, langsomme og ude af stand til at kommunikere med hinanden. Perspektivet er snarere aktuelt og fremadrettet. Ideen til skriftet blev fostret af Hasse Clausen og Martin Zachariasen som en udløber af undervisningen på faget ”Datalogiens videnskabsteori”. På dette selvreflekterende kursus, hvor bl.a. datalogiens påvirkning af det omgivende samfund drøftes, savnede vi de formidlende og perspektiverende bidrag fra forskere i datalogi.

Kendskabet til datalogisk forskning, og mere generelt it-forskning, er ikke videre udbredt i det danske samfund. Den manglende viden om, hvad it-forskning egentlig er, kan spores hele vejen op gennem uddannelsessystemet og anses af mange for at være en væsentlig barriere for at få flere unge til at tage en it-uddannelse. Gymnasielærere forespørger regelmæssigt de datalogiske institutter om relevant undervisningsmateriale, der kan styrke elevernes it-kompetencer. De gymnasiale uddannelser er således en særlig relevant målgruppe for formidling af aktuel forskning om it.

Formålet med jubilæumsskriftet er derfor ikke mindst at formidle aktuel datalogisk forskning til et bredt publikum og interesserende gymnasieelever. Sigtet er at beskrive forskningens centrale rolle i den informationsteknologiske udvikling. Redaktionen har inviteret elleve forskere med tilknytning til DIKU til at bidrage til skriftet. Bidragene spænder fagligt vidt, og de enkeltstående bidrag kan læses uafhængigt af hinanden.

Redaktionen har bestræbt sig på at sikre både faglig bredde og en formidlingsmæssig sammenhæng. Meget passende indledes skriftet derfor med et historisk radioforedrag af DIKUs første professor Peter Naur fra 1967 om faget datalogi. Ud over dette visionære og stadig højaktuelle bidrag har redaktionen placeret de enkelte artikler under fire tematiske indledninger, der giver en guidet rundtur i datalogiens univers: Den digitale verden; den tekniske platform; den digitale tekniks muligheder; menneskers omgang med den digitale teknik.

En række sponsorer har bidraget til, at dette skrift har kunnet opnå et tilfredsstillende kvalitetsmæssigt layout og kan udgives i bogform i et første oplag på 1000 eksemplarer. Listen af sponsorer kan ses bagest i værket.

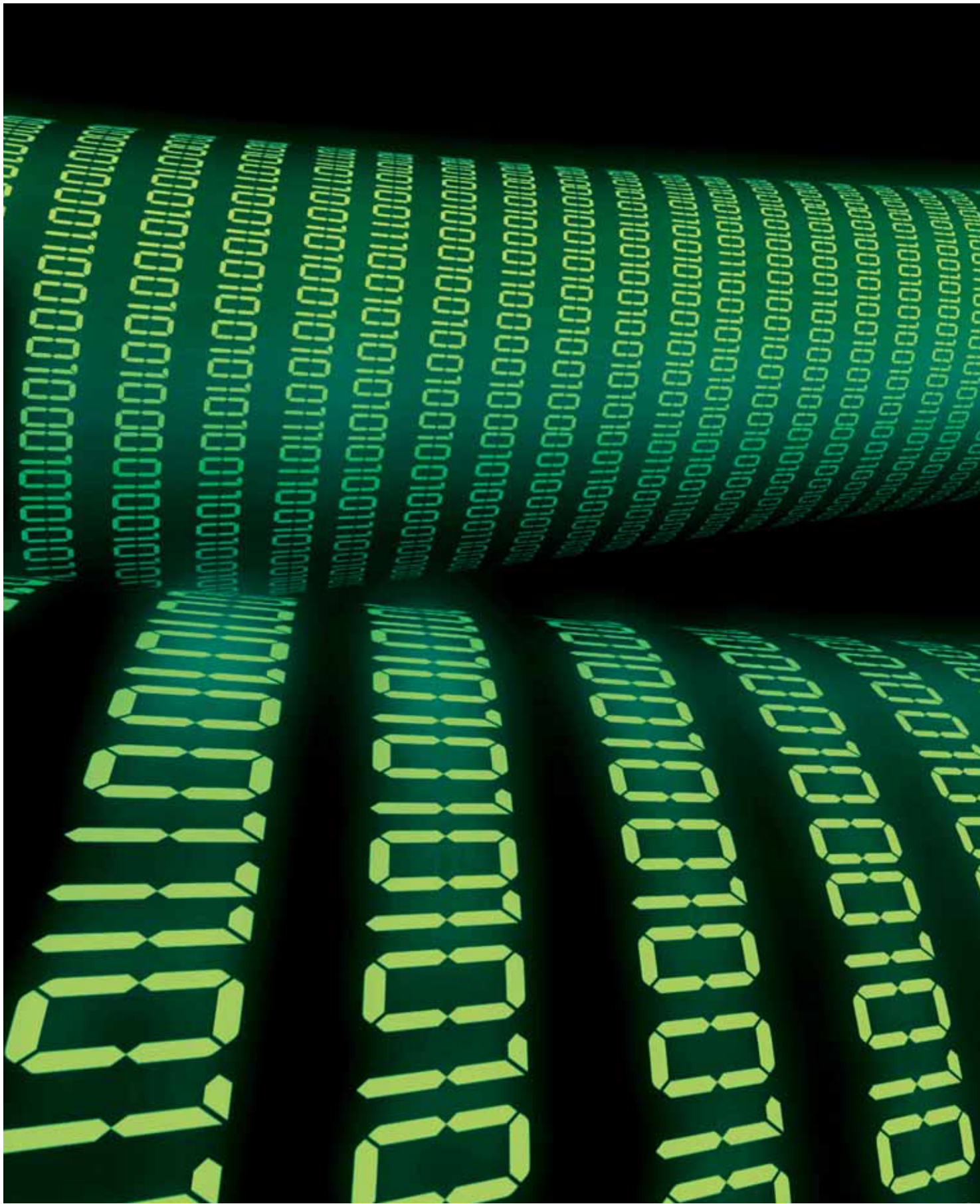


Redaktionen retter hermed en stor tak til de forfattere, som har bidraget med engagerede og inspirerende bidrag til skriftet: Peter Naur, Brian Vinter, Klaus Hansen, Torben Mogensen, Kenny Erleben, David Pisinger, Mads Nielsen, Morten Kringelbach, Esben Warming Pedersen, Peter Blume og Rasmus Helles.

Forfatterne har udvist en stor forståelse for skriftets overordnende ramme og har indgået i en meget konstruktiv dialog med redaktionen i bearbejdelsen af bidragene. Vi takker desuden Lisa Schultz for hendes bidrag til Peter Blumes artikel. Endelig takker vi forfatterne for deres tilladelse til, at skriftet gøres tilgængeligt for offentligheden på DIKU's hjemmeside: diku.dk/jubilee under en Creative Commons-licens. ♦

Redaktion:

Tariq Andersen
Jørgen Bansler
Hasse Clausen
Inge Hviid Jensen
Martin Zachariasen





DEN DIGITALE VERDEN - ET BARN AF OPLYSNINGSTIDEN





Den digitale verden – et barn af oplysningstiden

.....
Af **redaktionen**
.....

Gennem de sidste par årtier er en digital revolution fejet ind over vores tidligere så analoge samfund. Den elektroniske computer er blevet hvermandseje. Mennesker i alle aldre bruger dagligt computere til alverdens formål. Børn og unge surfer på nettet, chatter, spiller computerspil og kan ikke leve uden mobiltelefonen; for voksne er computeren uundværlig såvel på arbejdet som i hjemmet, og også pensionister søger nu oplysninger på nettet.

Penge er stort set erstattet af betalingskort og digitale transaktioner; i stedet for fotoalbum lægges billeder i dag på computeren; hvor musikelskere tidligere gik i pladeforretningen og købte gramfonplader eller cd'er, downloades musik i dag fra nettet; trykte vejkort er udkonkurreret af digitale navigationssystemer; papiraviser kæmper en hård kamp for at overleve i konkurrencen med netaviser. Og sådan kunne man blive ved med at opremse nye fænomener, der er produkter af den digitale tidsalder, og som godt nok ikke ændrer menneskers tilværelse radikalt, men indebærer stadige og mærkbare fremskridt for det enkelte menneske.

Det kan synes som de små skridts forandringer. Verden af i morgen adskiller sig ikke væsentligt fra verden af i dag, men anlægger vi et mere langsigtet perspektiv, tegner der sig et noget andet billede. Verden af i dag er ganske forskellig fra verden for 20 år siden, og verden om 20 år vil være helt anderledes end den verden, vi kender i dag.

Selvom udviklingen af mange kan opfattes som hæsblæsende, og det kan være vanskeligt at se, hvorledes den enkelte selv bidrager til denne udvikling, er det dog uomgængeligt, at udviklingen skabes af mennesker. De teknikere og specialister, der udvikler de nye digitale produkter, har selvsagt en fremtrædende rolle, ligesom alle de mennesker, der med stor begejstring bruger de nye produkter via de valg, de gør, har betydning for udviklingen.

Men også de mange mennesker, som stiller krav til politikerne om forbedringer i samfundet, fx om indførelse af regler og systemer for at opnå en lige og retfærdig behandling af mennesker, spiller en væsentlig rolle. Har vi først udtrykt ønsker om og krav til, at bestemte fænomener skal kunne måles, har vi lagt kimen til udviklingen af endnu et digitalt produkt. Dette gælder, hvad enten der er tale om CO₂-kvoter, test i folkeskolen, kontrol med SOSU-assistenters arbejdsindsats eller smileys i fødevarebranchen.

Den naturvidenskabelige tænkemåde er rammen

Menneskets trang til at måle alverdens ting samt sætte naturen i system er en væsentlig forklaring på fremvæksten af den digitale verden og også en medvirkende årsag til videnskabernes enestående succes igennem de seneste århundreder. Ikke alene har det haft betydning for vores forståelse af den

verden, vi lever i, men har også ført til et utal af nye erkendelser og opfindelser. Den naturvidenskabelige tænkemåde er blevet en del af almindelige menneskers tankegods.

Det centrale i den naturvidenskabelige tænkemåde er, at man opnår erkendelse om fænomener i verden ved at observere dem og beskrive dem i form af systematiske data, som man er i stand til at måle. Dette gjorde kendte videnskabsmænd som Tycho Brahe (1546-1601), Alexander von Humboldt (1769-1859) og Charles Darwin (1809-1882), og det var også sådanne tanker, der lå til grund for udgivelsen af den store franske encyklopædi (1751-1780, Denis Diderot (1713-1784)), hvor man søgte at bringe orden på verden og sætte naturen i system. Det er denne tankegang, vi genfinder bag de digitale systemer, vi anvender i dag, ud fra filosofien, at har vi først etableret en systematisk orden, kan vi styre og kontrollere fænomenet.

Den elektroniske computer er redskabet

Den naturvidenskabelige tænkemåde er den ene væsentlige faktor i udviklingen og etableringen af den digitale verden. Den anden væsentlige faktor er opfindelsen af den elektroniske computer. Den elektroniske computer (ENIAC blev som den første konstrueret af det amerikanske militær i 1945) er et redskab, som er meget mere end de mekaniske regnemaskiner, som Blaise Pascal (1623-1662) og Gottfried Wilhelm von Leibniz (1646-1716) opfandt. Det er en generel maskine, der kan håndtere alle former for data – ikke blot tal. Med computeren fik mennesket rådighed over et redskab, som på eminent vis understøtter den naturvidenskabelige tænkemåde. Såfremt vi er i stand til at beskrive et fænomen i form af data, så det fremtræder i en systematisk orden, kan vi benytte computeren til at håndtere denne beskrivelse af fænomenet. Vi kan lagre beskrivelsen, vi kan bearbejde beskrivelsen, vi kan transmittere og formidle beskrivelsen, alt sammen med en hastighed, som overstiger den menneskelige fatteevne. Forudsætningen er imidlertid, at fænomenet er beskrevet på en systematisk og ordnet måde.

Det kritiske spørgsmål i forhold til udnyttelsen af computeren og opbygningen af den digitale verden er, om vi kan give systematiske og ordnede beskrivelser i form af data, som giver mening. Vi kan selvfølgelig godt udforme en beskrivelse, så computeren kan lægge alle personnumre sammen, men giver det mening? Vi kan modtage alle verdens nyheder i en lind strøm, men giver det mening? Vi kan udstyre alle med en GPS, der informerer alle i verden om, hvor man er, men giver det mening? Der er uendelig mange anvendelsesmuligheder, men hvilke skal vi vælge? Det er blandt andet sådanne overvejelser, man skal gøre sig, når man anvender computeren som et teknisk redskab.

Vi har valgt at indlede dette jubilæumsskrift med Peter Naurs radiotransmitterede foredrag fra 1967, hvor han gør sig overvejelser om benævnelsen på og indholdet af det nye fag, som skulle oprettes på Københavns Universitet i lyset af computerens fremkomst. Man skal bemærke, at foredraget er fra et tidspunkt, hvor man godt nok havde teoretisk indsigt i computerens muligheder, men meget lidt eller stort set ingen praktisk erfaring, som vi kender det i dag. Man skrev sit program i hånden. Man var mange brugere, som anvendte den samme computer. Man kendte intet til skærme og mus, internettet eller smartphones. Arbejdet med computere var for specialister, og almindelige mennesker havde stort set ingen berøring med eller kendskab til computere. På trods heraf finder vi, at artiklen giver en fremragende beskrivelse af fagets grundlag og de muligheder, der ligger i faget datalogi – læren om data. ♦

Datalogi – læren om data

.....
Den anden af fem forelæsninger af **Peter Naur**, professor emeritus fra DIKU, der opstod som Danmarks Radios Rosenkjærforelæsninger i 1966. Forelæsningerne blev holdt i april og maj 1967 og herefter udgivet under titlen Datamaskinerne og samfundet af Munksgaard, København 1967
.....

I den første af disse forelæsninger stillede jeg Dem i udsigt at besvare spørgsmålet: Hvorfor er de elektroniske datamaskiner, eller datamaterne, som jeg foretrækker at kalde dem, af så væsentlig samfundsmæssig betydning? Jeg gennemgik kort regnemaskinernes historie og skitse-rede den vældige udvikling de har undergået i de sidste årtier, den udvikling der blandt andet har medført at vi i dag betragter disse apparater som datamanipulatorer, og ikke i første række som regnemaskiner.

Data, datarepræsentationer og dataprocesser

I dag skal vi beskæftige os med nogle helt andre grunde til datamaternes enestående betydning. Jeg vil skitsere det der ligger i begreberne *data* og de dertil nøje knyttede *datarepræsentationer* og *dataprocesser*, og forsøge at vise at disse begreber i sig rummer et helt nyt syn på en mangfoldighed af menneskets ytringsformer, både matematikken og mere dagligdags foreteelser som sproget. Gennem deres intime forbindelse med disse almenmenneskelige ting bliver datamaternes samfundsmæssige betydning langt mere forståelig.

For nu med det samme at få det værste overstået, her er den internationale definition af ordet ”data”:

DATA: enhver repræsentation af fakta eller ideer på en formaliseret måde, som kan kommunikeres eller manipuleres ved en eller anden proces.

Denne definition lyder formentlig ret så tam, eller endda tom, men just deri ligger det uhyre omfattende i begrebet. For at give Dem en mere håndgribelig forståelse vil jeg nu gennemgå nogle eksempler og sammenholde dem med definitionen.

Lad os begynde med sædvanlig sprogbrug. Som eksempel har jeg her på papiret foran mig skrevet en sætning. Sætningen læser:

”På væggen hænger en sollysende høstscene”.

Disse tegn på papiret foran mig er ifølge definitionen data. Lad os for at forvisse os herom først overveje den første del af definitionen, der siger: ”Enhver repræsentation af fakta eller ideer”. Her er der tale om forholdet mellem sætningen på papiret og visse andre omstændigheder i verden, maleriet, væggen, det at maleriet hænger der, og tanken om maleriets motiv. Det er dette forhold mellem sætningen og virkeligheden, der udtrykkes ved at disse data repræsenterer fakta og ideer. Den anden del af definitionen siger at repræsentationen må være formaliseret på en sådan måde

”

*Jeg vil ...
forsøge at
vise, at disse
begreber i sig
rummer et
helt nyt syn
på en mang-
foldighed af
menneskets
ytringsformer*

at den kan kommunikeres eller manipuleres ved en eller anden proces. Hvad angår den nævnte proces da kan jeg uden videre nævne en sådan, nemlig den der består i at jeg lader mit blik følge teksten og læser den op for Dem som jeg netop gjorde før. Derigennem har jeg kommunikeret disse data til Dem. Tilbage står kravet om at repræsentationen skal være formaliseret. Her tænkes der på den omstændighed at en tekst som den omtalte består af enkelte tegn, bogstaverne, der holder sig inden for et meget begrænset sæt af former, nemlig de henved tredive bogstaver i alfabetet, og ydermere at små uregelmæssigheder i det enkelte tegns fremtræden på papiret er uden betydning. Denne omstændighed er betingelsen for at man kan tale om en perfekt kommunikation af teksten. Hvis for eksempel en af mine tilhørere har nedskrevet sætningen så kan vi gå ud fra at blot han er fortrolig med dansk retskrivning så vil hans version af sætningen være aldeles den samme som min, bogstav for bogstav.

Definitionen af data gemmer dog på endnu mere, gennem det den *ikke* siger. Den udtaler sig aldeles ikke om nogen bestemt måde data kan eksistere og lader derved alle muligheder stå åbne. Allerede den anførte illustration giver en række eksempler herpå. De primære data findes som tryksværte på papir her foran mig. Alene derigennem at jeg har læst sætningen højt og at den er blevet overført til lytterne gennem radioen er disse samme fakta og ideer blevet repræsenteret på en række andre måder, først som en serie lyde jeg udtaler, derefter som elektriske signaler af flere forskellige former, før den omsider igen omsættes til lyd i Deres højttaler. Hver af disse former kan opfattes som data der repræsenterer de samme fakta og ideer.

Endnu en væsentlig ting skal bemærkes ved den givne definition af data, nemlig den udtrykkelige tale om *processer*. En ting eller begivenhed er ikke i sig selv data, men bliver det først når den indgår i en proces hvori dens repræsentation af fakta og ideer er det afgørende. Hvis jeg brænder et beskrevet stykke papir så underkastes skrifttegnene en forbrændingsproces, men dette gør dem ikke i sig selv til data. Dette er derimod tilfældet hvis en person læser dem. Vi ser heraf at databegrebet beskæftiger sig med en måde som mennesker forholder sig på til visse fænomener.

Kunst og modeller som data

Databegrebet ligger nær op ad symbol- og modelbegreberne. Ordet symbol bruges både om matematiske symboler, hvor det knap kan skelnes fra data, og i kunsten. Brugen af symboler i kunsten er dog en kompliceret sag. Nok er der tale om at én ting på en ejendommelig måde peger på, eller henviser til, en anden, men der gives ikke nødvendigvis et krav om en formalisering som ved databegrebet. Således er der forskel mellem et maleri og data. Maleriet er ikke formaliseret og ingen kopi af det vil regnes for lige så god som originalen.

At formelle processer ikke er kunsten uvedkommende fremgår på den anden side af forbindelsen mellem litteraturen og bogtrykkkunsten, som må betragtes som en tidlig triumf for mekanisk databehandling. Ved at fremdrage dette eksempel håber jeg også at berolige dem der føler sig uhyggeligt til mode ved de moderne datamater. Vi har alle for længst vænnet os til at omgås særdeles personligt med data som er frembragt gennem mekaniske processer. Der er ingen der føler sig stødt af at kærlighedsdigte er reproduceret ved hjælp af ganske følelsesløse trykkemaskiner. Af og til kan man dog iagttage at der måske er en grænse for hvor vidt vi er parat til at acceptere

dette upersonlige mellemled. Jeg selv føler mig snydt når jeg modtager en julehilsen der er trykt og hvor afsenderen ikke engang har umaget sig til at tilføje sin egenhændige underskrift. Fornemmelsen kan dog også vendes om. Nutildags regnes et festtelegram jo for en fuldt ud passende form for hyldest, næsten bedre end et brev, til trods for at telegramteksten som bekendt har været udsat for en langt mere omfattende håndtering af uvedkommende personer og maskiner.

Databegrebets forbindelse til modelbegrebet er måske ikke ved første øjekast indlysende, da man ved en model normalt tænker på en ting som i sin fremtoning har stærk lighed med det den er en model af, som for eksempel ved modelfly. Modelbegrebet er imidlertid for længst overtaget i de empiriske videnskaber, fysikken, astronomien, biologien, og mange andre, og bruges der om langt mere abstrakte konstruktioner. I astronomien taler man for eksempel om stjernemodeller, der består af kurver eller tabeller, altså data.

Data som værktøj

Men hvis databegrebet på alle disse områder er foregrebet eller indeholdt i disse kendte begreber, hvad bringer det da af nyt, må vi spørge. Svaret herpå ligger allerede i det sagte, på den ene side i den ubegrænsede frihed til som data at vælge hvilken art fænomener som helst, lyde, tryksværte på papir, elektriske strømme, magnetfelter, osv., dels i interessen for processerne. I denne sidste interesse ligger noget aktivt. Data er ikke noget der eksisterer eller som man har, det er noget man bruger, et værktøj for mennesker i deres virksomhed.

Denne opfattelse kan umiddelbart bekræftes ved en overvejelse af sproget. For så vidt sproget anvendes til kommunikation af fakta og ideer mellem mennesker, og ikke som medium for fri fabuleringen, da tjener sproget åbenbart som hjælpemiddel, værktøj, - og er vel endda det vigtigste menneskelige værktøj overhovedet.

Men med en helt fri opfattelse af data åbner vi vejen for en langt mere effektiv udnyttelse af dem. Sagen er at den lethed hvormed vi kan gennemføre en given dataproces i høj grad afhænger af præcis hvilke data vi har valgt til at repræsentere virkeligheden, eller som jeg kort vil sige, af *datarepræsentationen*. Dette forhold er velkendt fra sædvanlig brug af de sproglige repræsentationer, tale og skrift. Blandt de processer der her er tale om kan nævnes adskillige, for eksempel: at udveksle ideer blandt personer; give en fuldstændig beskrivelse af et forhold; at registrere nu tilgængelige fakta for at undgå at de fortabes; at udbrede en enkelt persons ideer til mange. Som enhver ved vælger man sin sproglige form, tale eller skrift, under hensyn til hvorledes processen vil forløbe mest effektivt. Til at udveksle ideer bruger man helst samtale, fordi den skriftlige form er for tung og langsom. Fuldstændige beskrivelser udformes bedst skriftligt, fordi dette er mere fordelagtigt for den der skal bruge dem. Til at registrere tilgængelige fakta bruger mange af os skriftlige notater, men det afhænger af vor indbyggede procesformåen og nogle af os opnår en sikrere registrering ved at bruge hukommelsen. Til at udbrede en enkelt persons ideer til mange benytter man i ikke helt simple tilfælde en kombination af tale og skrift, som vi for eksempel stadig kan iagttage det i arbejdet for at udbrede politiske ideer.

Den helt fri opfattelse af data har i praksis for længst slået sig igennem hvad angår den dataproces der hedder kommunikation. Vi benytter frit en blanding af de gammelkendte repræsen-

tationer, tale og skrift, og en mangfoldighed af andre, telegrafi, telefoni, radio- og fjernsyns-
overføring, idet vi i hver livssituation vurderer den indbyrdes fordel af disse i forhold til den
kommunikationsproces vi har brug for at få udført.

På området datalagring, som kan opfattes som den proces at kommunikere med en forsinkelse
af ubekendt varighed, har en voksende frigørelse fra sædvanlig nedskrivning ligeledes fundet
sted med fremkomsten af båndoptagere.

Fra datalogiens synspunkt må denne udvikling opfattes blot som en begyndelse, da den kun berø-
rer de allersimpleste dataprocesser. Fremkomsten af datamaterne betyder at langt mere kompli-
cerede dataprocesser meget effektivt kan udføres uden at mennesker behøver at deltage. Vejen er
derved blevet åbnet for at arbejdet med datamodeller bliver i høj grad automatiseret og
effektiviseret.

”
*Datamodel-
lerne tillader
at vi eksper-
menterer med
virkeligheden*

Datamodeller i samfundslivet

Betydningen af datamodeller for samfundslivet kan næppe overvurderes. De er grundlaget for
enhver planlægning, fra forretningsmandens tabeller og kurver over den erhvervsmæssige udvik-
ling, over ingeniørens arbejde med beregninger og tegninger der beskriver den bro han er ved at
konstruere, til fysikerens matematiske teori for atomkernen. På hvert af disse områder tillader
arbejdet med data at man skaffer sig viden om hvordan den virkelige verden vil opføre sig under
forskellige omstændigheder. Datamodellerne tillader at vi eksperimenterer med virkeligheden,
men uden at vi behøver opføre bekostelige eksperimentopstillinger og uden at tabe ret meget hvis
eksperimentet mislykkes. Værdien heraf er især slående ved planlægningen af ting som skibe,
flyvemaskiner, broer, og atomreaktorer, hvor et fejlslagent virkeligt eksperiment øjensynlig kan
være katastrofalt. Derimod betyder det ikke alverden om det ved en beregning viser sig at en
påtænkt reaktorkonstruktion vil eksplodere.

Hvad er nu betydningen af det datalogiske synspunkt for denne brug af modeller? Her er der to
ting at nævne. For det første befrier datalogien os for fordomme om at arbejdet med en bestemt,
given problemstilling er knyttet til en bestemt datarepræsentation. Sådanne fordomme er uhyre
udbredte. For eksempel arbejder ingeniører og arkitekter traditionelt med tegninger af deres kon-
struktioner som primære data. Det vil umiddelbart vække modstand hvis man foreslår at denne
form måske slet ikke bør være enerådende. Ikke desto mindre er det en kendsgerning at der nu til
dags i betydeligt omfang foregår udviklingen af komplicerede konstruktioner, som vejanlæg og
skibe, uden nævneværdig brug af tegninger.

Hermed er ikke sagt at den hidtidige brug af tegninger har været en datalogisk fejltagelse. Hvad
datalogien siger, er, at datarepræsentationen må afpasses efter den transformation der skal udfø-
res og det procesværktøj der er til rådighed. Indtil de moderne datamaters fremkomst omkring
1945 var arkitekter og ingeniører i det væsentlige henvist til at arbejde kun med mennesker som
procesværktøj. I relation til den måde vi mennesker arbejder med hovedet og til vor synsmekanik
er tegninger en særdeles velvalgt datarepræsentation.

Den anden værdi af datalogien for modelbyggeren ligger i selve opbygningen af en datamodel og i beskrivelsen af alle detaljer i de processer der indgår i dens brug. Den nøje formulering af en datamodel virker i høj grad stimulerende for forståelsen af enhver art problem, stimulerende på en måde som må opleves for at forstås. Denne dybere forståelse af problemerne er slet ikke knyttet kun til mere komplicerede problemer. En betydelig del af gevinsten kan også høstes uden at man kommer i berøring med datamater. Det afgørende er oplevelsen af hvordan dataprocesser kan udvikles, beskrives og bearbejdes.

Det fremgår af disse betragtninger at datalogien ikke har nogen nødvendig forbindelse med datamaterne og at den datalogiske betragtningsmåde indeholder meget gammelkendt, som blot ikke hidtil har været samlet under samme synspunkt. Datamaterne har dog det med sagen at gøre at de ved deres fremkomst fremtvinger en ny og velfunderet datalogisk stillingtagen til problemernes løsning. Den hurtighed, sikkerhed og prisbillighed hvormed nutidens datamater gennemfører komplicerede dataprocesser betyder et så afgørende spring sammenlignet med hvad der hidtil har været kendt, at enhver menneskelig aktivitet der i nævneværdig grad har brug for dataprocesser må blive påvirket deraf. Vanskeligheden ved overgangen beror i vid udstrækning på at dyrkerne af de mange specialfag slet ikke er vant til at se deres fag på den led som situationen kræver. Den ingeniør der bygger broer er vant til at vurdere konsekvenserne af nye materialer og nye konstruktive ideer, men ikke i nær så høj grad af nye beskrivelsesformer, nye datapræsentationer. Samtidig griber en ændring i den datarepræsentation der benyttes under projekteringen så dybt ind i arbejdet at den ikke kan overlades til fremmede eksperter alene, men kræver meget aktiv medvirken af det enkelte fags dyrkere.

Dertil kommer at datamaterne selv undergår en udvikling, således at tilpasningen af de enkelte fags problemer til dette værktøj ingenlunde er noget der kan klares én gang for alle.

Datalogi i uddannelsen

Har man således indset at datalogien på den ene side sammenfatter en lang række centrale menneskelige aktiviteter og begrebsdannelser under ét samlende synspunkt, og på den anden side formår at befrugte og forny tankegangen i en lige så lang række fag, da kan man ikke være i tvivl om at datalogien må have en plads i almenuddannelsen. For at nå til en rimelig forestilling om hvordan denne placering bør være er det naturligt at sammenligne med fag af lignende karakter. Man vil da nå frem til sproglære og matematik, som er de nærmeste analoge. Både datalogien og disse to fagområder beskæftiger sig med tegn og symboler der er opfundet af mennesker som hjælpemidler. Fælles for de tre emner er også deres karakter af redskaber for mange andre fag. I uddannelsen må de derfor indgå på to måder, dels som hjælpefag ved studier af mange andre fag, dels som hovedfag ved uddannelsen af specialister i selve disse emner. Vi har jo alle måttet gennemgå meget betydelige mængder sprog, regning og matematik i skolen, uanset at kun ganske få af os er blevet lingvister eller matematikere. På lignende måde må datalogien bringes ind i skoleundervisningen og forberede os alle på i tilværelsen i datamaternes tidsalder, ganske som læsning og skrivning anses som en nødvendig forudsætning for tilværelsen i et samfund der er præget af tryksager.

Datalogien kan tænkes placeret i skoleundervisningen, enten som et selvstændigt fag, eller som en del af matematikken. Det afgørende er indholdet af undervisningen. Hovedtemaerne må være

data, datarepræsentationer, og dataprocesser. Disse fundamentale begreber må belyses gennem en række konkrete eksempler som bekvemt kan tages fra områder inden for skoleelevernes erfaringsområde, talregning, stavning, opslag i registre, og lignende. Der bør også indgå simple eksempler på brugen af mekaniske og elektriske fænomener til datarepræsentation. Disse eksempler bør belyses ved simple forsøg. Datamaterne bør også omtales, men ikke som det centrale i faget, snarere som en afsluttende orientering.

Når datalogien er blevet vel etableret i almenuddannelsen vil den mystik der omgiver datamaterne i manges forestillinger opløse sig i intet. Dette må betragtes som den måske allervigtigste begrundelse for at fremme forståelsen af datalogien. Det vil nemlig være betingelsen for at herredømmet over datamaterne og deres anvendelse ikke bliver en sag for en lille gruppe af eksperter, men bliver en sædvanlig politisk sag, og således gennem det politiske system kommer til at ligge hvor det bør, hos os alle. ❖

”
Når datalogien er blevet vel etableret i almenuddannelsen, vil den mystik der omgiver datamaterne i manges forestillinger opløse sig i intet

PETER NAUR



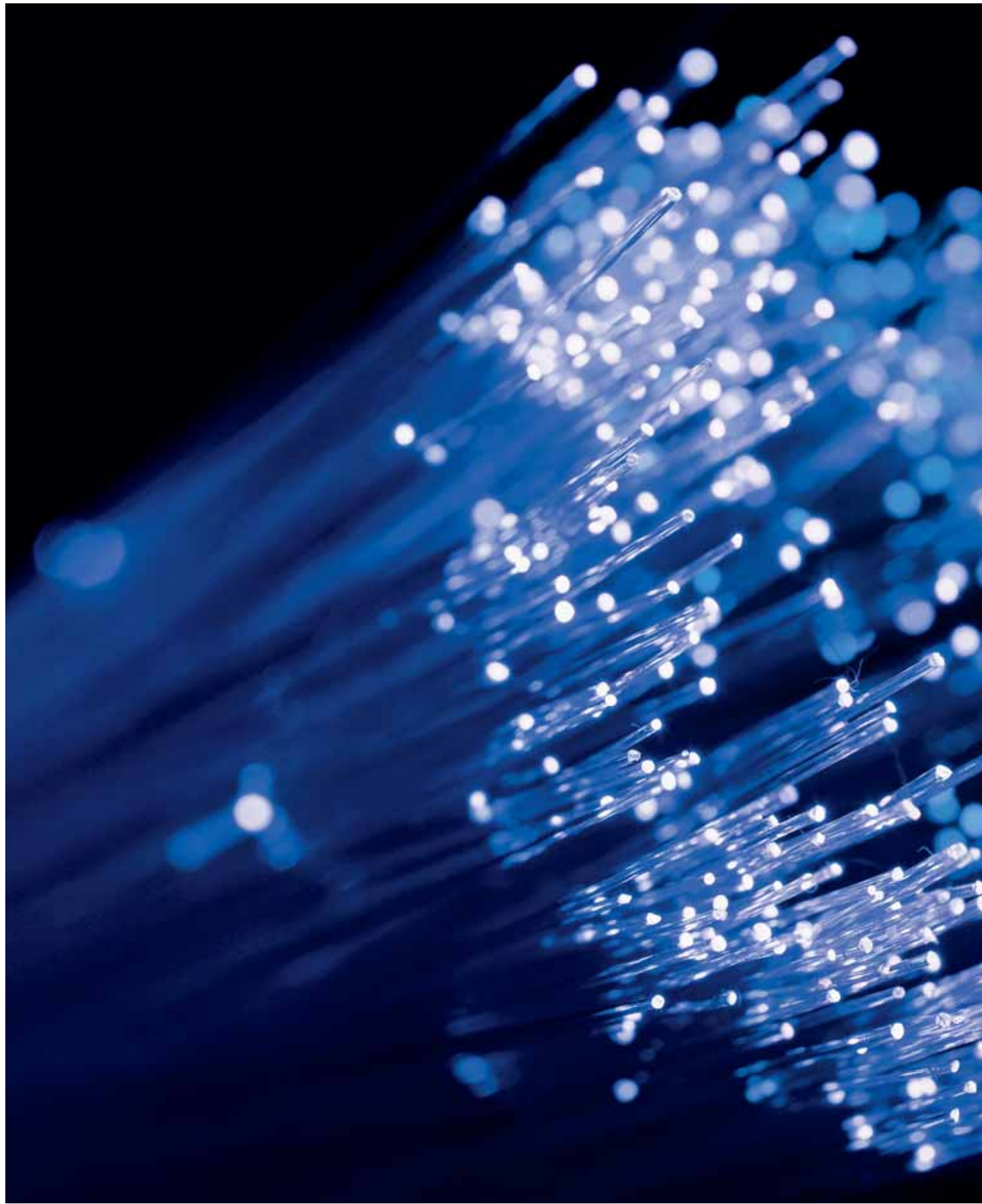
Peter Naur har været professor emeritus fra DIKU siden 1998. Han er uddannet astronom og erhvervede doktorgraden fra Københavns Universitet i 1957. Snart efter sin ansættelse ved Regnecentralen i 1959 slog han sit navn fast i den internationalt gryende aktivitet omkring datamaskinerne ved sine bidrag til

det internationale programmeringssprog ALGOL 60. I de følgende år førte hans overvejelser over Regnecentralens ALGOL-kompilatorer til, at han i 1966 formulerede en akademisk disciplin under betegnelsen datalogi, overvejelser som i 1972 blev publiceret under titlen "Concise Survey of Computer Methods".

Dette førte til, at Københavns Universitet i 1970 oprettede Datalogisk Institut, hvortil han blev kaldt som professor. Som sådan fortsatte han i de følgende år sine studier over programmeringens problemer med stigende opmærksomhed på at forstå den menneskelige tankevirksomhed i alle dens manifestationer. En lang række af

hans arbejder på dette område blev i 1992 udgivet samlet under titlen "Computing – A Human Activity". Hans fortsatte studier førte derefter til udgivelsen af en række bøger: "Knowing and the Mystique of Logic and Rules" (1995), "Antifilosofisk leksikon: Tænkning – sproglighed – videnskabelighed" (1999), "Antiphilosophical Dictionary: Thinking – Speech – Science/Scholarship" (2001), "Psykologi i videnskabelig rekonstruktion" (2002), "An Anatomy of Human Mental Life – Psychology in Unideological Reconstruction – Incorporating the Synapse-State Theory of Mental Life" (2005) og "The Neural Embodiment of Mental Life by the Synapse-State Theory" (2008). I disse bøger fremlægges en lang række originale indsigter om menneskelig tænkning og den måde, tænkning foregår på i menneskers hjerner og nervesystemer.

Peter Naur blev tildelt G.A. Hagemann Medaillen i 1963, Jens Rosenkjær Prisen i 1966 og Computer Pioneer Award of the IEEE Computer Society i 1986. I 2005 blev han tildelt computerverdens mest prestigefyldte hæderspris, Association for Computing Machinerys A. M. Turing Award.





DEN TEKNISKE PLATFORM





Den tekniske platform

.....
Af **redaktionen**
.....

Computeren repræsenterer en teknologi, som er tæt knyttet til den naturvidenskabelige tilgang. Teknologisk udvikling går således hånd i hånd med videnskabelig udvikling. Dette er en relativt ny situation, idet teknologisk udvikling i tidligere tider var tæt knyttet til håndværket og det praktiske arbejde, mens videnskabeligt arbejde var forbeholdt privilegerede og lærde personer, som ikke nødvendigvis beskæftigede sig med de praktiske anvendelser af deres teoretiske overvejelser.

Dette forhold er forstærket radikalt med fremkomsten af computeren. Videnskabsfolk eksperimenterer og observerer, udformer ordnede og systematiske beskrivelser, fortolker symboler og hændelser, bearbejder og omformer foreliggende beskrivelser, formulerer modeller og teorier, som er udformet i symboler. Kort sagt: Videnskabsfolk er informationsarbejdere – eller mere præcist symbolarbejdere. Med computeren har de fået et særdeles kraftfuldt redskab, som de kan anvende i deres videnskabelige arbejde.

Computeren kan håndtere og bearbejde alle mulige former for symboler, herunder opbygge modeller af forskellige fysiske såvel som tænkte fænomener. Man kan lave vej- og klimamodeller, økonomiske modeller, trafikmodeller, modeller af hjernen, ja stort set modeller af alt, som man formår at beskrive på systematisk og ordnet vis. Alle disse modeller indgår som en del af den digitale verden – ikke nødvendigvis som nøjagtige kopier af de betragtede fænomener, men som forenklede beskrivelser, som man kan bearbejde og eksperimentere med som en nyttig del af den videnskabelige forskning og udvikling.

Turingmaskinen og von Neumann-arkitekturen

Årsagen til, at computeren er så kraftfuldt et redskab, er, at alle computere i princippet er udformet i henhold til to teoretiske modeller: Turingmaskinen og von Neumann-arkitekturen.

Alan Turing (1912-1954) udformede i 1936 en matematisk model for en simpel symbolmanipulerende maskine, som trods sin enkle opbygning kunne udføre ikke-trivielle beregninger og simulere logikken i enhver computer. At vi kan relatere enhver computer til samme teoretiske grundmodel, er baggrunden for, at vi i princippet kunne sammenkoble alle verdens computere og i teoretisk forstand opfatte den samlede mængde af computere som én computer. At dette så ikke altid kan lade sig gøre i praksis, er en anden sag, men som teoretisk ideal spiller det en væsentlig rolle for opbygningen af den digitale verden.

John von Neumann (1903-1957) har fået sit navn uløseligt forbundet med det grundlæggende princip, som er gældende for arkitekturen i computeren. Den altafgørende idé er, at man kan

betragte det program (det sæt af instruktioner), som skal behandle nogle data, som selv værende data. Man kan således lagre det skrevne program i computeren som data, og dermed kan computeren bearbejde det lagrede program. Det er denne grundlæggende betragtning, der er baggrunden for, at man kan udvikle oversættere til programmeringssprog, og at man kan udnytte computeren til at oversætte en beskrivelse i et programmeringssprog til en tilsvarende beskrivelse i et andet programmeringssprog. Man kan altså udnytte og udbygge tidligere systemer til nye systemer, oven i købet uendelig mange gange. Dette er forklaringen på, at vi ser fremkomsten af stadig nye versioner af allerede velkendte og velfungerende it-produkter.

Man udbygger og forbedrer de eksisterende systemer til nye, som kan lidt mere end de gamle versioner. Man opbygger lidt efter lidt en teknologisk platform, som dernæst vil være fundamentet for en ny udvikling. På denne vis får man skabt stadig mere avancerede anvendelser af teknologien, som er brugbare for almindelige mennesker, uden at de skal bekymre sig om eller tænke på den komplekse teknik, som udgør grundlaget for den umiddelbart anvendelige teknologi. Almindelige mennesker kan således i dag benytte internettet til alle mulige former for kommunikation på samme simple vis, som de kan udnytte elektriciteten ved at tænde på kontakten.

Udvikling og konstruktion af it-produkter

Én ting er, at vi har et teoretisk grundlag, som vi kan henvise til, når vi vil forstå og forklare, hvorledes det er muligt at opbygge den digitale verden. Noget andet er, at udvikling og konstruktion af de konkrete it-produkter er praktisk arbejde, hvor de ideale betingelser, som anlægges i teorien, ikke er gældende.

I praksis er man underlagt en lang række begrænsninger. Det kan være manglende lagerplads, begrænset processorhastighed eller utilstrækkelige programmeringssprog. Dette betyder, at man inden for it-branchen arbejder med at forbedre teknikken, eksempelvis ved at kreere hurtigere og mindre computere, og den enkelte programmør, der skal vælge, hvilket programmeringsredskab han skal anvende i en konkret situation, må overveje, hvorvidt han ønsker at opfylde de teoretiske idealer eller de praktiske mål. På denne måde er arbejdet med at konstruere nye it-systemer nok så meget et praktisk håndværk som en videnskabelig disciplin, og ofte ligger der et stort pionerarbejde til grund for etableringen af nye teknologiske platforme. Når der er formuleret en idé om en teknisk løsning, uden at man egentlig ved, hvordan denne skal udformes, vil der ofte være mange mennesker rundtomkring i verden, der uafhængigt af hinanden vil søge at skabe en konkret løsning. Eksempler på dette er idéen om, at mennesker skulle kunne programmere computere i et mere dagligdags sprog, idéen om, at hvert menneske skulle have sin egen personlige computer, samt idéen om, at alle computere skulle kunne forbindes med hinanden i et verdensomspændende netværk.

Pionerarbejdet er en væsentlig del af den teknologiske udvikling og har sit eget særlige liv. Det vil ofte være ”nørder” med stor indsigt i det tekniske, som udfører dette arbejde. De er enspændere, som finder sammen med ligesindede og etablerer lokale og internationale miljøer, hvor de kan udveksle erfaringer. Det er et kreativt og hektisk miljø, hvor der fantaseres, eksperimenteres og diskuteres indtil den dag, hvor der etableres en alment accepteret løsning, hvorefter pioner-

arbejdet stille og roligt dør ud, og arbejdet med videreudvikling af den teknologiske platform overlades til den etablerede del af branchen.

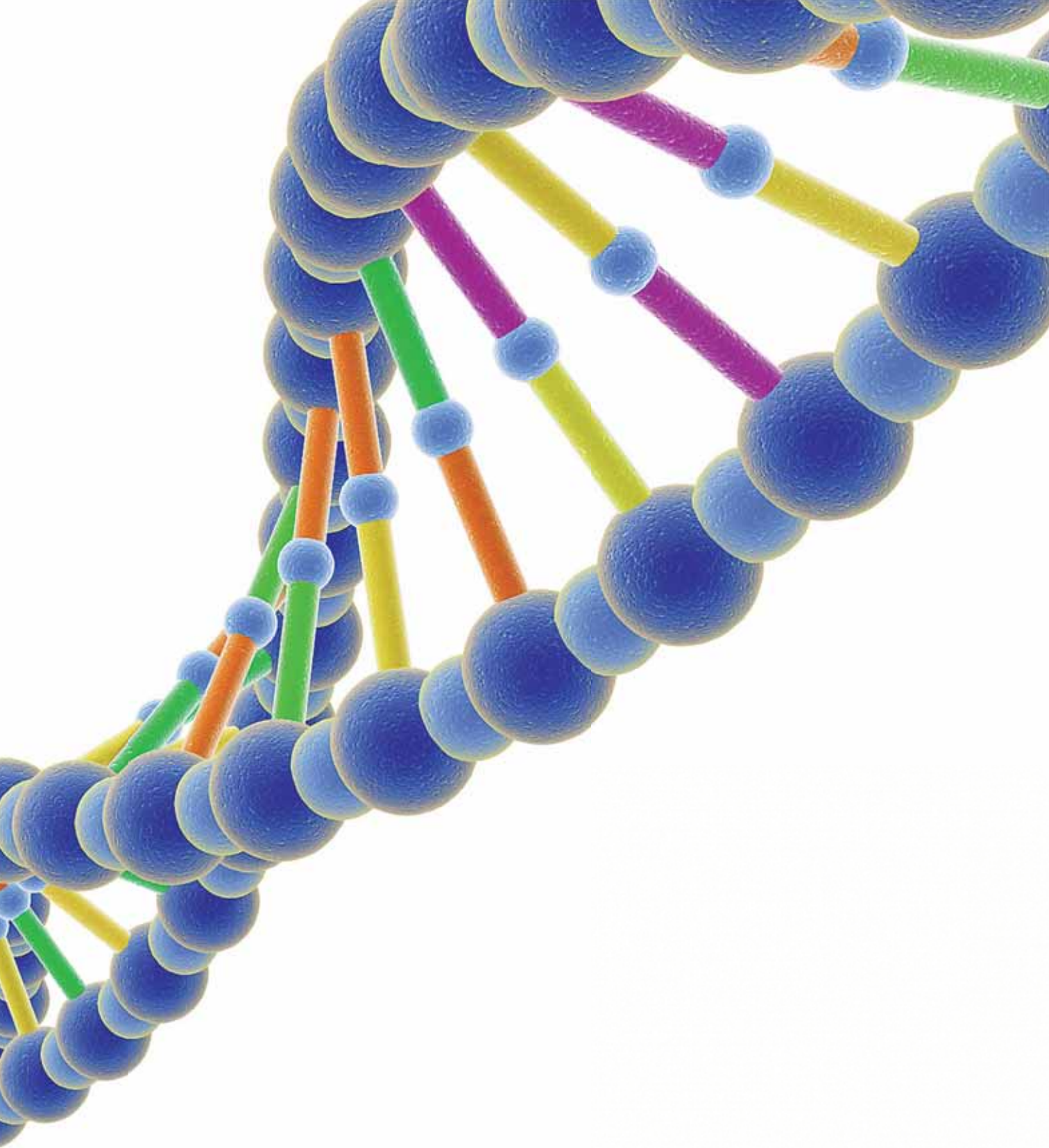
I det følgende beskrives i tre artikler, hvordan konkrete maskiner og programmer, hvis eksistens vi i dag betragter som en selvfølge, er opstået som følge af ihærdige pionerånders utrættelige indsats:

Brian Vinter fortæller med eksempler om udfordringerne med at konstruere supercomputere, som har kapacitet til at udføre beregningsopgaver af astronomiske dimensioner.

Klaus Hansen giver en personlig beretning om, hvordan forløberen for internettet, nemlig de akademiske net, opstod og udviklede sig i danske og udenlandske universitetskredse.

En væsentlig forudsætning for den digitale udvikling er fremkomsten af de forskellige programmeringssprog. Torben Mogensen fortæller om de allerførste pionerer på området og om udviklingen frem til nutidens moderne programmeringssprog. ♦





Supercomputere – mange bække små...

.....
Af **Brian Vinter**, DIKU
.....

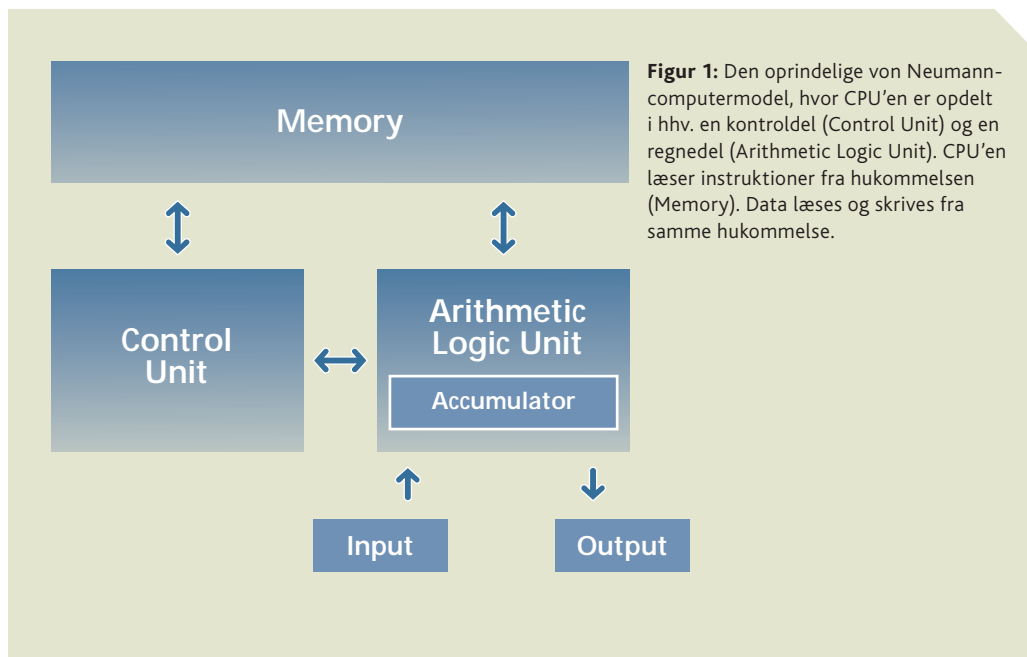
Hastigheden på en computers centrale regneenhed, CPU'en, har altid været en vigtig konkurrenceparameter for leverandørerne. Med hurtigere computere får vi bedre vejrudsigter, og vi kan simulere processer i fysik og kemi, som vi ellers måtte bygge kostbare eksperimenter for at gennemskue. Når man søger på Google, leder deres computere i 100 PB (100.000.000.000.000 bytes) af information, som de har kopieret fra internettet. For at det skal kunne ske så hurtigt, som det gør, må de mange data være gennemregnet på forhånd – så samtidig med at informationen på internettet stiger, stiger behovet for hurtigere computere tilsvarende. Tidligere beroede computerens hastighed på taktfrekvens, og vi så hastigheden på en pc gå fra 5 MHz til 4 GHz fra 1979 til 2004, dvs. 800 gange hurtigere taktfrekvens på blot 25 år. I dag markedsføres CPU'er med flere og flere kerner, hvilket løst kan oversættes til "Køb 1 CPU, og få 2, 4, 8 eller endnu flere". Men begejstringen over disse mange-kerne-CPU'er er til at overskue, hvorfor?

Computerens taktfrekvens

Computerens hastighed angives ofte som dens taktfrekvens. CPU'en fungerer på den måde, at den har et lille ur, der slår takten for dens fremgang; hver gang uret slår, kan CPU'en begynde en ny instruktion. Når en computer har en 3 GHz CPU, betyder det altså, at CPU'ens ur slår 3 milliarder gange i sekundet.

Den grundlæggende computermodel

Helt basalt fungerer en computer ved, at den centrale regneenhed (Central Processing Unit eller CPU'en) udfører instruktionerne i et program en ad gangen, se figur 1. Instruktionerne til programmet ligger i computerens hukommelse, og det samme gør de data, som programmet regner på. Den mest udbredte computermodel hedder von Neumann-modellen, og her ligger både instruktionerne til programmet og de data, computeren regner på, i den samme hukommelse. I von Neumann-modellen består det, vi i dag tænker på som en CPU, af to dele, dels en kontroldel, der styrer udførelsen af et program, dels en regnedel. Der findes også en model, hvor instruktioner og data ligger i hver sin hukommelse, kaldet Harvard-modellen, men den ses sjældent anvendt i generelle computere.



I dag har en almindelig computer ofte to eller flere CPU-kerner, hvilket betyder, at den har flere CPU-lignende enheder, som deler en fælles hukommelse og derfor skal samarbejde for at løse en beregningsopgave.

Moderne forskning kræver mange FLOPS

Måleenheden for en computers regnekraft hedder FLOPS, floating point operations per second, eller på dansk: antal beregninger på kommatalt pr. sekund. Vi har brug for meget stor regnekraft for at lave moderne videnskab, hvor computere bruges til at beregne kemiske bindinger i mediciner, styrken af og elektriske egenskaber ved nanopartikler, effekten af en ny vindmølle eller jordens fremtid i form af klimamodeller – bare for at nævne nogle få anvendelser, som kræver meget store supercomputere.

Verdens hurtigste kommatalt-computer i år 2010 har en teoretisk hastighed på 2,3 billioner FLOPS, også skrevet som 2,3 PFLOPS, og en målt hastighedsrekord på 1,76 PFLOPS. Denne computer kan altså lave 1,76 billioner beregninger pr. sekund.

Men det er ikke kun verdens supercomputere, der bliver stadig hurtigere. Det samme gælder almindelige pc'er, som de fleste mennesker i dag har mindst én af. I dag er fire eller otte CPU-kerner og mere end 30 GFLOPS ganske almindeligt. Faktisk har en pc, der er beregnet til spil, en speciel Graphics Processing Unit (GPU) til at beregne grafik. Denne GPU har ofte 240 kerner og regner med så meget som 1,4 TFLOPS.

”

*Måleenheden
for en compu-
ters regnekraft
hedder FLOPS*

FLOPS

Floating point **O**perations **P**er **S**ekund eller på dansk: Antal beregninger på kommatall pr. sekund. En beregning kan fx være en addition, en subtraktion, en multiplikation eller en division af to kommatall.

1 FLOPS = 1 beregning pr. sekund
1 MFLOPS = 1 million, 10^6 beregninger pr. sekund
1 GFLOPS = 1 milliard, 10^9 beregninger pr. sekund
1 TFLOPS = 1 billion, 10^{12} beregninger pr. sekund
1 PFLOPS = 1 billion, 10^{15} beregninger pr. sekund

Definitionen af FLOPS forudsætter, at de kommatall, der benyttes, kun har et begrænset antal cifre, dvs. at det samlede antal cifre før og efter kommaet er begrænset. Der findes to typer af FLOPS, single precision (SP) og double precision (DP). Et DP-kommatall har dobbelt så mange mulige cifre som et SP-kommatall. En DP er dobbelt så stor som en SP, og når FLOPS bruges som måleenhed, er det altid DP, man bruger, medmindre man skriver, at det er SP. Oftest er en CPU's SP-ydelse dobbelt så stor som dens DP-ydelse.

Verdens hurtigste supercomputer

Den til enhver tid hurtigste computer i verden kan findes på www.top500.org.

Lige nu (år 2010) er det den amerikanske computer "Jaguar", der topper listen. Denne computer har

224.162 kerner, der tilsammen kan levere en faktisk regnekraft på 1,76 PFLOPS, mens den teoretiske hastighed er 2,3 PFLOPS, altså lige over 10 GFLOPS pr. kerne.



*Den til enhver
tid hurtigste
computer i verden
kan findes på
www.top500.org*



Moore's lov forudsiger udviklingstakten i CPU'ers hastighed

At vi kan bygge stadig hurtigere computere, skyldes en teknologisk udvikling, der er formuleret i Moore's lov, som ikke er en lov i naturvidenskabelig forstand, men snarere en slags tommelfingerregel, som siger, at størrelsen på en transistor på et integreret kredsløb som fx en CPU halveres hver 18. måned.

Tidligere forudsagde Moore's lov også, at CPU'ernes taktfrekvens blev fordoblet hver 18. måned; det skyldtes, at transistorerne jo var blevet halvt så store, og derfor kunne computeren skifte tilstand på den halve tid. Desværre medførte dette, at CPU'erne også brugte stadig mere strøm, hvilket igen betød, at de blev hurtigere ophedet.

Da Intel i 2004 annoncerede, at de ikke ville bygge en 4 GHz Pentium 4 CPU, bekendtgjorde de samtidig, at hvis de skulle fortsætte som hidtil, ville en CPU i 2007 have 16 GHz taktfrekvens og blive 6.000 grader varm, mens den kørte. Derfor har vi ikke siden 2004 set nogen nævneværdig stigning i taktfrekvensen på en CPU, og computeren bliver derfor ikke længere hurtigere til at køre programmer på en enkelt CPU-kerne.

Men Moore's lov gælder stadig. Det betyder, at en CPU, der i dag har 500 millioner transistorer, om 1½ år vil have 1 milliard transistorer og om 3 år 2 milliarder transistorer. For at lave en kommatal-beregning, fx addition af to tal, skal vi bruge ca. 1.000 transistorer, så en CPU med 1 milliard transistorer kunne, hvis den udelukkende skulle addere, foretage 1 million beregninger pr. taktslag. En hurtig CPU har i dag næsten 4 milliarder taktslag i sekundet, så den ville altså kunne lave næsten 4 millioner milliarder eller 4 billioner beregninger pr. sekund (4 PFLOPS). Nu skal en CPU jo ikke kun addere men også kunne lave mange andre ting, som der skal bruges transistorer til. Et rimeligt gæt ville være, at en milliard transistorer skulle regne med 1 TFLOPS. Desværre er virkeligheden, at vi bruger ganske få af transistorerne på en CPU til at regne med. En almindelig CPU med en milliard transistorer leverer snarere 100 GFLOPS, mens en GPU kommer tæt på 1 TFLOPS med 1 milliard transistorer.

Hvorfor kan man ikke bare skalere op?

Moore's lov forudsiger, at en computer får stadig mere regnekraft, for mens taktfrekvensen ikke stiger, så får hver computer stadig flere kerner. Hvis en kerne kan regne med 10 GFLOPS, så burde 1.000 kerner vel kunne regne med 10 TFLOPS? Desværre er det ikke så let, idet man for at kunne udnytte 1.000 kerner må omskrive programmet til at kunne deles op i mindre dele, som så kan køre samtidig – man taler om at parallelisere arbejdet.

Det lugter af et simpelt regnestykke fra folkeskolen: Hvis én mand kan grave en grøft på en time, hvor hurtigt kan to mænd så gøre det? En halv time ville vi nok hurtigt gætte på, men hvad med 1.000 mænd? Kan de gøre det på 3,6 sekunder? Det er klart for enhver, at man ikke kan grave en grøft på 3,6 sekunder – de mange mennesker ville simpelthen stå i vejen for hinanden. Sådan er det også, når vi skal parallelisere computerprogrammer; det kan være meget vanskeligt, selv for de letteste problemer.

Lad os prøve et meget simpelt forsøg. Vi vil gerne finde summen af en masse tal, helt præcist 10.240 tal. Lad os sige, at vi har en computer med 1.024 kerner, dvs. at hver kerne skal finde sum-



*Moore's
lov gælder
stadig*

men af ti tal. Dette kan gøres på ni taktslag ved at lægge de to første tal sammen, lægge summen til det tredje tal til osv. Derefter har vi 1.024 delsummer, og dem må vi så lægge sammen. Hvis en kerne gør det, så vil det tage 1.023 taktslag, og det er jo meget mere, end vi brugte på at få opgaven ned fra 10.240 til de 1.024 tal, der skal summeres. De første 90% af opgaven vil altså gå meget hurtigere end de sidste 10%, så det må vi kunne gøre bedre!

Løsningen er selvfølgelig at dele opgaven op, således at 512 kerner lægger tallene sammen to og to, derefter 256, 128, 64, 32, 16, 8, 4, 2 og til sidst 1 kerne for den sidste sum. Det betyder, at vi må bruge yderligere ti taktslag til at løse opgaven. Det betyder, at vi bruger dobbelt så lang tid på at løse opgaven, som man skulle tro, når man dividerer opgavens størrelse med antallet af kerner i vores computer. Dette er naturligvis bare overfladen af problemet. At lægge tal sammen er den letteste opgave, vi kan forestille os, og den lader sig jo forholdsvis let parallelisere. Der er mange andre og mere værdifulde opgaver, der er langt vanskeligere at parallelisere.

Hukommelsen er også for langsom

Desværre er problemet med de mange kerner ikke den eneste udfordring, som dataloger står over for, når computere skal levere mere regnekraft. Det helt store problem ligger i selve von Neumann-modellen, nemlig i, at CPU'en hele tiden skal læse fra hukommelsen – og desværre er hukommelsen meget langsommere end CPU'en. Hukommelsen i en computer er opbygget i en langsommere teknologi for at kunne presse så meget hukommelse ind som muligt. En almindelig pc har måske 4 GB af typen dynamisk hukommelse. Hvis man skulle bruge en hurtigere type, kaldet statisk hukommelse, kunne den samme computer kun have plads til ca. 256 MB hukommelse eller 16 gange mindre hukommelse. En hurtig hukommelse kan CPU'en læse fra på 15 nanosekunder, altså 15 milliardtedele af et sekund. Men en hurtig CPU-kerne kører ved 3 GHz, dvs. at den har en taktfrekvens på 3 milliarder instruktioner pr. sekund. Det betyder så, at en kerne har tre taktslag hvert nanosekund, men hukommelsen kan altså kun tilgås hvert 15. nanosekund. Med andre ord er en CPU-kerne 45 gange hurtigere end den hukommelse, den skal læse fra. Faktisk er CPU-kernen opbygget sådan, at den kan køre fire instruktioner for hvert taktslag, så CPU-kernen er faktisk 180 gange hurtigere end hukommelsen. Hvis vi har 12 kerner på en CPU, er den altså $12 \times 180 = 2.160$ gange hurtigere end hukommelsen. Hukommelsen er 2.160 gange langsommere, og en CPU skal både læse instruktioner og data fra hukommelsen, så vi har altså et enormt problem med den langsomme hukommelse.

Den klassiske løsning på problemet med den langsomme hukommelse hedder cachehukommelse. Idéen med cachehukommelse er at have en hurtig, men relativt lille hukommelse, der ligger meget tæt på CPU'en, og så have en kopi af en del af den langsomme hukommelse liggende i den hurtige hukommelse. Problemet med cachehukommelsen er, at den kun virker, når computeren kan gætte, hvilke data CPU'en skal bruge, eller hvis programmøren har skrevet programmet på en måde, så man ved, at CPU'en kan gætte, hvilken del af hukommelsen der skal bruges. Moderne CPU'er er udstyret med ganske meget cachehukommelse i et forsøg på at forhindre CPU-kernerne i at vente hele tiden, og en stadig større del af de transistorer, vi kan bruge i en CPU, bliver brugt til cachehukommelse. Det kan synes som lidt af et ressourcespild at bruge gode transistorer på noget så "kedeligt" som at kopiere hukommelsen. Derfor er der også en række nye tendenser i CPU-design, som vi vil se på sidst i artiklen.

Hvor meget hurtigere kan computeren blive?

Når vi skal bygge meget store supercomputere, må vi bruge mange CPU-kerner. De største computere har mere end 200.000 CPU-kerner, og selvom elektronikdelene er ganske små, fylder 200.000 CPU'er stadig temmelig meget, ofte så meget som en hel sportshal. CPU'erne må kommunikere over ganske lange afstande, og der har vi det næste problem: lysets hastighed. CPU'erne kommunikerer med strøm via et netværk, og strøm bevæger sig noget langsommere end lys (afhængigt af det materiale, netværket er lavet af).

For at gøre det let vil vi i dette afsnit antage, at computere kan snakke sammen med lysets hastighed. Det lyder temmelig fjollet at sige, at lyset bevæger sig langsomt – vi siger jo netop ”med lysets fart”, når noget går stærkt. Men i denne sammenhæng er lyset faktisk ret langsomt. Lysets hastighed er ca. 300.000 km pr. sekund, det samme som 1 fod eller 30 cm pr. nanosekund. Vi har tidligere regnet ud, at en CPU-kerne ved 3 GHz og 4 instruktioner pr. taktslag kan udføre $3 \cdot 4 = 12$ instruktioner pr. nanosekund. Dvs. at mens lyset bevæger sig 1 m, kan en sådan CPU udføre 40 beregninger på den samme tid. Hvis vi går ud fra, at en computer er 50 m lang, kan en CPU altså udføre 4.000 instruktioner på den tid, det tager at sende en besked fra den ene ende af computeren til den anden og så få et svar tilbage. Den faktiske tid er pga. elektronikken omkring 1.000 gange langsommere. Hvis en CPU skal bede en anden CPU gøre noget af arbejdet for sig, skal det arbejde altså være på mindst 4 millioner instruktioner – ellers bruger den mere tid på at kommunikere end den tid, som CPU'en sparer ved ikke at gøre arbejdet selv. Disse tal er alle for én computer i ét lokale; hvis man i stedet vil bruge computere, der er forbundet via internettet, til at samarbejde, så er det hele igen en million gange langsommere. Kun rigtig store opgaver kan sendes til en anden computer, hvis man skal spare tid på det.



I denne sammenhæng er lyset faktisk ret langsomt

Forhold mellem afstand og antal CPU-instruktioner

I tabellen herunder antages det, at vi kan kommunikere med lysets hastighed, og at vi har en 3 GHz CPU, der udfører 4 instruktioner pr. taktslag.

Afstand	Tid for lyset	CPU-instruktioner
1 fod	1 nanosekund	12
1 m	3,3 nanosekunder	40
1 km	0,0000033 sekunder	40.000
Jorden rundt ved ækvator (40.066 km)	0,13 sekunder	1.602.640.000

Tre tendenser i CPU-design

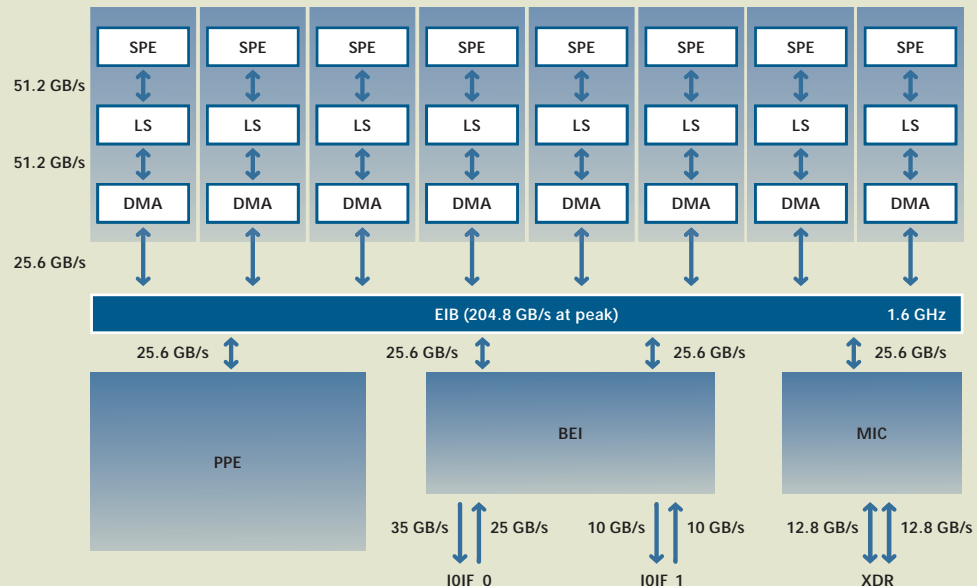
Dataloger og ingeniører står over for en stor udfordring. Moores lov forudsiger, at vi får stadig flere transistorer på en CPU; det giver flere kerner, der kan lave beregninger samtidig. Derudover er hukommelsen langsom, og når CPU'er skal tale sammen, er selv lysets hastighed langsom. Problemet med den langsomme hukommelse er det mest umiddelbare og derfor det problem, der arbejdes hårdest på at løse. Der synes at være tre generelle tilgange til problemet:

1. Gøre hukommelsen hurtigere
2. Gøre hukommelsen hurtigere i gennemsnit
3. Gøre CPU'en relativt langsommere.

IBM har med Cell-BE-CPU'en, som man bl.a. finder i PlayStation 3-konsollen, arbejdet med at gøre hukommelsen hurtigere. I stedet for at bruge den klassiske cachehukommelse, hvor CPU'en forsøger at gætte, hvilke data programmet skal bruge, så har CELL-BE+ en lille, men hurtig hukommelse til hver CPU-kerne. Denne hukommelse er så den eneste, som kernen kan bruge, men kernen har derudover en lille hjælpekomponent, der kan overføre blokke af hukommelse mellem den lille, hurtige og den store, langsomme hukommelse.

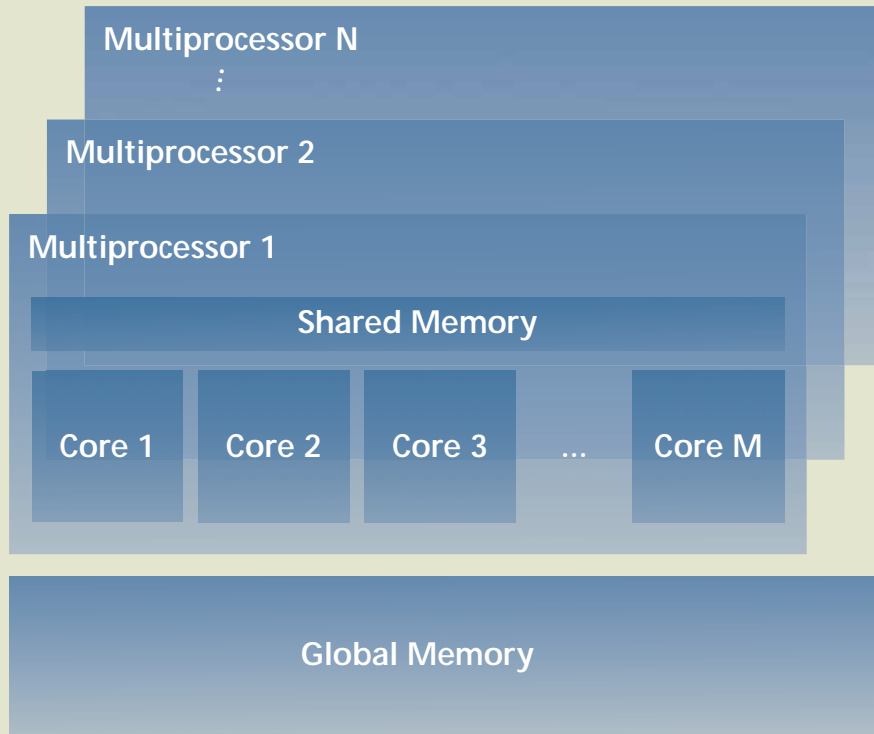
IBM CELL-BE+

CELL-BE+ CPU'en har ni kerner: én klassisk CPU-kerne og otte specielle CPU-kerner med hver deres hukommelse. De ni CPU-kerner er forbundet af et internt, meget hurtigt netværk. Hver enkelt CPU kan regne med mere end 100 GFLOPS. Spillekonsollen PlayStation 3 bruger en ældre og mindre version af den samme CPU, CELL-BE.



NVidia Fermi

Tesla, som er den største processor fra NVidia lige nu (2010), fås med 448 kerner, der kan levere 515 GFLOPS regnekraft. NVidia har tidligere fokuseret på grafikprocessorer, og Tesla er den første CPU, de udvikler, der er tiltænkt videnskabelige beregninger.



Firmaet NVidia, som er bedst kendt for at lave grafikprocessorer, GPU'er, er også gået ind på markedet for videnskabelige beregninger. Deres løsning er at gøre hukommelsen hurtigere i gennemsnit. Idéen er lidt den samme som at bruge en lastbil til at flytte med frem for en personbil. Lastbilen kan ikke køre hurtigere end personbilen – ofte omvendt – men til gengæld kan den have meget mere med. I en computer svarer dette til, at man frem for at nøjes med 8 bytes, når man læser fra hukommelsen, i stedet henter 448 bytes eller endnu mere. Dette fungerer selvfølgelig kun, når man skal bruge alle de bytes, der hentes. Det problem løser NVidia ved at dele de mange bytes, der hentes, op imellem mange, op til 448, CPU-kerner. Dermed bliver det op til programøren at skrive et program, der kan køre parallelt på så mange kerner, at man udnytter størrelsen på den hukommelse, der hentes.

Den sidste model, at gøre CPU'en relativt langsommere, virker måske fjollet, men når CPU'en er for hurtig til hukommelsen, er der kun to muligheder tilbage: at gøre hukommelsen hurtigere eller at gøre CPU'en langsommere. Sun bruger den sidste model, og idéen er meget simpel. Hvis programøren alligevel skal skrive et parallelt program, så kan programmet jo skrives til mange flere kerner, end der faktisk er på CPU'en. I Suns Niagara-processorlinje deler man en CPU-kerne op i 8 eller 16 virtuelle kerner over tid. Det fungerer på den måde, at hvis man bruger to virtuelle kerner, så vil hver af disse få hver andet taktslag, dvs. at CPU'en har fået dobbelt så mange kerner, der hver kører halvt så hurtigt. Det giver den samme totale beregningskraft, men til gengæld er hastighedsforskellen til hukommelsen for hver kerne blevet halvt så stor. Når man så gør det med 16 virtuelle kerner, bliver forskellen reduceret til 1/16; til gengæld skal en CPU med 8 fysiske kerner så programmeres, som om den havde 128 kerner.

Sun Niagara

Niagara-CPU'erne fra Sun, T1, T2 og T3, bruger virtuelle CPU-kerner til at få programmerne til at køre langsommere. Selvom det lyder ulogisk, fungerer det fint, fordi den samlede regnekraft i CPU'en ikke bliver mindre. Til gengæld virker hukommelsen ikke så langsom for de enkelte kerner.

Core 0.0	Core 1.0	Core 2.0	Core 3.0	Core 4.0	Core 5.0	Core 6.0	Core 7.0
Core 0.1	Core 1.1	Core 2.1	Core 3.1	Core 4.1	Core 5.1	Core 6.1	Core 7.1
Core 0.2	Core 1.2	Core 2.2	Core 3.2	Core 4.2	Core 5.2	Core 6.2	Core 7.2
Core 0.3	Core 1.3	Core 2.3	Core 3.3	Core 4.3	Core 5.3	Core 6.3	Core 7.3
Core 0.4	Core 1.4	Core 2.4	Core 3.4	Core 4.4	Core 5.4	Core 6.4	Core 7.4
Core 0.5	Core 1.5	Core 2.5	Core 3.5	Core 4.5	Core 5.5	Core 6.5	Core 7.5
Core 0.6	Core 1.6	Core 2.6	Core 3.6	Core 4.6	Core 5.6	Core 6.6	Core 7.6
Core 0.7	Core 1.7	Core 2.7	Core 3.7	Core 4.7	Core 5.7	Core 6.7	Core 7.7

Nogle friske forskningsresultater

For programmører er de flere forskellige tilgange til at løse hastighedsproblemet faktisk en dårlig nyhed. Det er ikke til at forudsige, hvilken teknologi der viser sig at være den bedste, og de forskellige løsninger kræver alle forskellige måder at programmere på. Det betyder, at hvis man som programmør beslutter sig for at skrive til modellen med de virtuelle kerner, og den teknologi så viser sig ikke at overleve, så kan programmerne ikke umiddelbart bruges til de næste generationer af CPU'er. Et af forskningsprojekterne på DIKU går ud på at løse dette problem og forsøger at udvikle en programmeringsmodel, der passer til alle tre løsninger.

Idéen er at tage udgangspunkt i et eksisterende programmeringssprog – i dette tilfælde Python – og så sprede opgaverne, som et program specificerer, ud over de CPU'er, der er til rådighed.

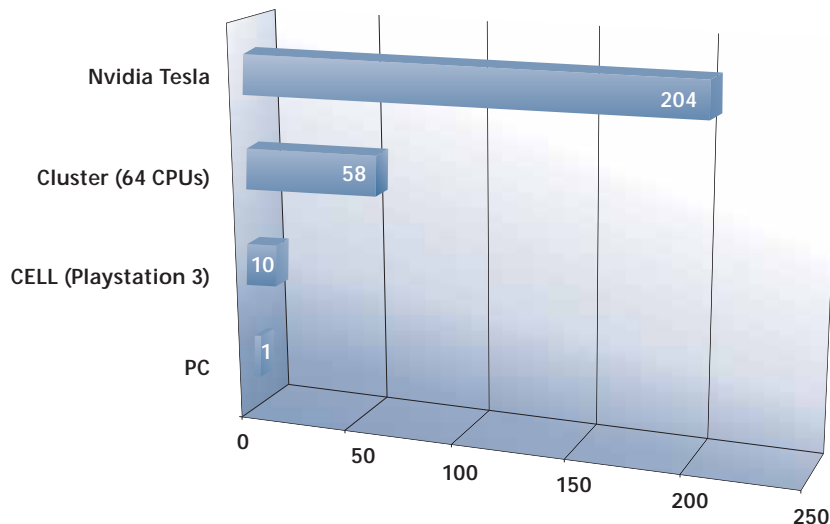
Projektet er i skrivende stund i sin opstart, men der er allerede kommet nogle resultater. Test-programmet i figur 2 er et meget simpelt program, der forsøger at beregne værdien af π med en metode, der kaldes Monte Carlo-simulering, og programmet er meget kort:

```
def McPi(s=samples):  
    x,y=(random(s),random(s))  
    (x,y)=(x*x,y*y)  
    x+=y  
    return sum(less_equal(x, 1.0))*4.0/s
```

Figur 2: Monte Carlo-simulering.

Denne type opgave er meget let at parallelisere, og man kan da også se på resultaterne i figur 3, at de er rigtig lovende. Tallene kaldes speed-up. Vi har altså en almindelig kraftig pc, der kører almindelig Python, som får basisnummeret 1. En PlayStation 3 er 10 gange hurtigere, en klynge-computer med 64 kerner er 58 gange hurtigere, og et Tesla-kort fra NVidia er 204 gange hurtigere end den almindelige pc. Alle fire maskiner kører præcis det samme program, så programmøren skal ikke ændre noget for at skifte teknologi.

Men det betyder også, at programmøren kan blive fri for at bekymre sig om, hvilken teknologi der i sidste ende viser sig at være den bedste, fordi programmer skrevet i DIKUs system kører effektivt på alle tre typer uden nogen form for ændringer.



Figur 3: Testresultater foretaget med Monte Carlo-metoden. Grafen viser den relative merydelse i forhold til en almindelig pc, som de parallelle maskiner kan præstere.

Supercomputeren på Københavns Universitet

Københavns Universitet råder over en supercomputer, der har 4.560 CPU-kerner og kan regne med en hastighed på 45 TFLOPS. Computeren anvendes af såvel eScience-centeret (et tværfagligt forskningscenter på KU, der tillige uddanner kandidater og ph.d'er i eScience) som flere af de individuelle institutter under Det Naturvidenskabelige Fakultet. Computeren anvendes til en række meget forskellige forskningsprojekter, der har det tilfælles, at de kræver ekstrem regnekapacitet.

Eksempelvis bruges supercomputeren af Niels Bohr Institutets forskere til at regne på de meget store mængder data, der kommer fra Large Hadron Collider-eksperimentet på CERN (i pressen bedre kendt som big bang-eksperimentet). Et andet anvendelsesområde er simulering af galakseres udvikling eller atomers bevægelser i helt små molekyler.

Derudover anvendes supercomputeren i stor stil inden for medicinsk forskning til at finde nye og bedre diagnosticeringsmetoder af en lang række sygdomme. Bl.a. kan man få computeren til at hjælpe med at stille tidlige diagnoser af sygdomme som slidgigt, KOL og brystkræft ved at analysere CT- og MR-billeder. Om anvendelse af supercomputeren i medicinsk forskning henvises desuden til Mads Nielsens artikel om machine learning. ❖

Læs mere

http://en.wikipedia.org/wiki/UltraSPARC_T3

[http://en.wikipedia.org/wiki/Cell_\(microprocessor\)](http://en.wikipedia.org/wiki/Cell_(microprocessor))

http://en.wikipedia.org/wiki/Nvidia_Fermi



Figur 4: Supercomputeren på Københavns Universitet – der driver en række forskningsprojekter på Det Naturvidenskabelige Fakultet.

BRIAN VINTER



Brian Vinter er professor på eScience-centeret, som han har været tilknyttet siden 2007. Han er civilingeniør fra Aalborg Universitet og ph.d. fra Tromsø Universitet fra 1999. Brian har en lang akademisk karriere, bl.a. som gæsteforsker på Princeton University, CERN i Schweiz og University of Kent i England.

Han har været aktiv inden for klyngecomputer-forskning siden 1994 og har ydet betydelige bidrag til forskningen i feltet, specielt inden for distribueret delt hukommelse. Han er på DIKU også kendt som Mr. Supercomputer. Brian Vinter underviser på DIKU i fagene ekstrem multi-programmering og cluster computing. Ud af en lang liste af publikationer kan nævnes: "Next Generation Processors" (2009) og "Cycle-Scavenging in Grid Computing" (2009).

Internetpionerer i Danmark

.....
Af **Klaus Hansen**, DIKU
.....

Forløberen for internettet var de akademiske net, som var en række regionale kommunikationsnet med udspring i universitetsmiljøerne, og som opstod nogenlunde samtidigt i flere lande rundt omkring i verden.

Denne beretning illustrerer, at den udvikling, der gav os internettet i Danmark i starten af 1990'erne, var resultatet af en mangeårig og snørklet proces, der bestemt ikke har fulgt en snorlige projektplan, men ofte var præget af tilfældighedernes spil og af bestemte menneskers evner, holdninger og indgriben på et givet tidspunkt. Den viser også, at det i udviklingsprocessen ikke alene drejede sig om at få teknik og økonomi på plads, men i høj grad også om at kunne samarbejde i grupper, der var meget broget sammensat på tværs af landegrænser og fagområder, i en søgen efter fælles beslutninger. Forskelle i sprog, kultur, faglige mål og ikke mindst den faglige baggrund og de arbejdsmæssige traditioner dannede indimellem barrierer, der tog tid og en større indsats at få brudt ned.

Dette bidrag er baseret på personlige erfaringer og oplevelser som ansat ved Regnecentralen og ved DIKU. Starttidspunktet er valgt ud fra Centernettets tidlige dage i sidste halvdel af 70'erne. Jeg begyndte at arbejde med datakommunikation i 1974, og på dette tidspunkt havde der inden for det offentlige allerede i nogle år været gang i forskellige aktiviteter, så tidspunktet er lidt vilkårligt valgt. Sluttidspunktet er valgt ud fra, at jeg skiftede forskningsområde, og at de kommercielle ISP'er (internet service provider) var kommet godt i gang.

Baggrund - situationen i 1970'erne

I 1975 så samfundet og dermed netverdenen meget anderledes ud end nu. Det danske samfunds kommunikation foregik hovedsagelig med brevpost og telefon. I universitetsmiljøet brugte man datakommunikation til at give brugerne af de store centrale maskiner adgang hertil fra deres arbejdspladser; DIKU fik en middelhastighedsterminal med hulkortlæser og langsom printer i 1971 med transmissionshastigheden 2.000 bit/sekund med adgang til den store IBM-computer i Lyngby. Der

En typisk RC3600 fra ca. 1976 med papirbåndslæser, konsolskrivemaskine, strimmelhuller og et kabinet med mini-computer, operatørpanel og båndstation.



var tre regionale centre, NEUCC, RECKU og RECAU, se faktaboksen om edb-centre, med hver sin maskintype, operativsystem og netteknologi. Rundtomkring på institutterne var der computere af meget forskellig størrelse og af ret forskellige fabrikater, og som typisk kun betjente lokale brugere.

Teknologisk var billedet meget broget, idet hvert computerfirma havde udviklet sin egen teknologi uden at skele til konkurrenterne. De services, se faktaboks om kommunikationsformer, der var til rådighed, omfattede stort set kun interaktiv terminaladgang og overførsel af data ved batch-overførsel. Flyselskaber og banker havde transaktionssystemer, men de var i alle henseender lukkede.

Regionale edb-centre

NEUCC blev etableret i 1965 som et edb-center placeret på DTH (nu DTU) i Lyngby, der skulle levere computerkapacitet til Norden og Nordtyskland.

På basis af erfaringerne med NEUCC blev der i 1969 oprettet tre regionale centre for forskning og uddannelse: NEUCC fortsatte som regionalt center, og to nye centre blev oprettet ved Københavns Universitet (RECKU) og Aarhus Universitet (RECAU). I 1985 blev de tre centre organisatorisk lagt sammen i Danmarks Edb Center for Forskning og Uddannelse kaldet UNI•C.

Edb-kapacitetsudvalget blev oprettet af Undervisningsministeriet til at forvalte de statslige bevillinger til edb inden for forskning og uddannelse. Disse gik i høj grad til de tre regionale centre, men også til mindre computere ved universiteter, institutter og skoler.

Kommunikationsformer

Batch Databehandling, hvor opgaver sendes til udførelse på et senere tidspunkt.

I forbindelse med net samler brugeren typisk et job bestående af programmer og data i form af et bundt hulkort og sender det fra en terminal til en central computer, hvor jobbet sættes i kø. På et tidspunkt afvikler computeren det og sender resultatet tilbage til terminalen, hvor det printes.

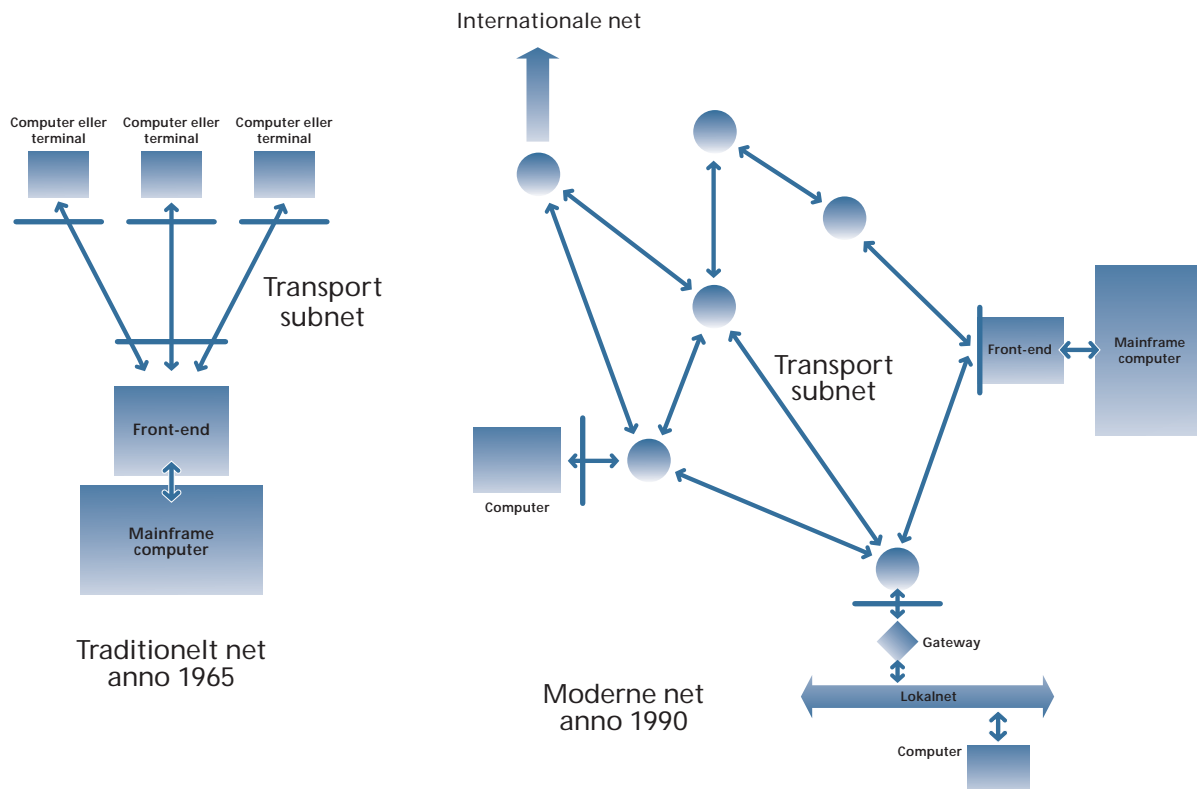
Under udførelsen af jobbet kan computeren arbejde med filer lagret på en lokal disk eller magnetbånd.

Interaktiv Databehandling, hvor brugeren sender kommandoer fra en terminals tastatur til computeren og får resultaterne med det samme, som udskrives på papir eller vises på en skærm. Interaktivt og online er synonymmer.

Person til person-kommunikation

Der er flere måder at kommunikere på mellem personer:

- 1) e-mail eller sms, hvor der sendes elektroniske beskeder,
- 2) news, hvor et indlæg distribueres til abonnenter, og
- 3) konferencesystemer, bulletin boards eller blogs, hvor brugere kan følge med i andres indlæg.



Figur 1: I 1965 var det almindeligt, at nettenes formål kun var at give brugere adgang til en central computer fra deres arbejdsplads; typiske hastigheder var 110/300 bit/s for interaktive terminaler og op til 2.400 bit/s for batch-terminaler. Undtagelser fra denne model var fx flyselskaberne, som arbejdede med transaktioner og havde mange samarbejdspartnere. Denne arkitektur ændres i løbet af 1970'erne, således at der indføres et transportnet, som transporterer data mellem vilkårlige computere, store som små. I løbet af 1980'erne sammenkobles små computere med lokalnet, og det er derefter gerne lokalnettene, der forbindes ved transportnettet. Det er derfor essentielt, at grænsefladen til transportnettet (den tykke linje) og transportprotokollene er standardiserede.

Arpanettet blev oprettet i 1969 og blev i 1983 opdelt i en militær del og en ikke-militær del. Den sidstnævnte blev senere en del af internettet. Erfaringerne fra Arpanet-projektet havde peget på, hvorledes fremtidig datakommunikation kunne se ud, se figur 1:

- En lagdelt arkitektur, hvor et fysisk subnet transporterer data uafhængigt af applikations-typen, og som har et eller flere standardiserede interfaces og dermed tillader, at alle de forskellige computerfabrikater kan kobles sammen
- Standarder for applikationsprotokoller, som tillader peer to peer-kommunikation (P2P) og klient/server-kommunikation mellem vilkårlige computere.

Med udgangspunkt i denne vision om åbne systemer (OSI eller open systems interconnection) var problemet nu:

- at få tilvejebragt de nødvendige standarder
- at få leverandørerne til at udvikle fysiske subnet
- at få leverandørerne til at programmere applikationer, der fulgte standarderne, og gerne på en måde, så programmet kunne bruges på mange forskellige platforme
- at få organiseret det nødvendige driftsorienterede samarbejde
- at uddanne specialister
- at få finansieret driften
- at informere beslutningstagerne (ikke mindst) om den fagre nye verden.

Dette var store opgaver, som kun kunne løses ved samarbejde på tværs af landegrænser.

Centernettet

Edb-kapacitetsudvalget, se faktaboks om edb-centre, som tog sig af at etablere den nødvendige computerkapacitet i den danske universitetsverden, var klar over, at der skulle styres mod at realisere åbne net for forskere, og igangsatte i 1977 et projekt ved navn Centernettet, som havde til formål at give forskere og undervisere adgang til alle danske computere, uanset brugernes geografiske placering, og at give adgang til de store udenlandske databaser.

To konsulenter fra Regnecentralen A/S, hvoraf jeg var den ene, udarbejdede sammen med folk fra de tre regionale centre udbudsmaterialet for Centernettet i maj 1978. Udbuddet omhandlede hardware og software til et transportnet og implementering af protokoller til interaktiv terminaladgang og overførsel af filer.

Standarder

Højre del af figur 1 illustrerer vigtigheden af, at udstyr, der kobles sammen, følger samme standarder. Der er dels tale om grænsefladen til transportnettet, dels om kommunikationen mellem to computere, der begge er koblet til transportnettet. Der findes en international standard kaldet OSI Referencemodellen, som gør det simpelt at tale om dette.

Internettet, som det ser ud i 2010, er baseret på brugen af protokoller, som tilhører TCP/IP-familien, men da disse egentlig kun omfatter transporten gennem subnettet og den overordnede kommunikation mellem to computere, bruges også standarder fra ISO (International Standardisation Organisation). De vigtigste standarder omfatter transport gennem subnettet ved protokollerne IP og TCP (fra 1977) og standarder for WWW (HTTP og HTML, 1992) og e-mail (1982). Arpanettet gik over til TCP/IP i 1983.

I 1978 var der ingen internationale standarder for transport og datakommunikation mellem computere, men til gengæld havde de store leverandører hvert sit bud på arkitektur og protokoller. Det britiske og det norske akademiske net havde ligeledes udviklet egne protokoller.

Organisationen CCITT, der standardiserede inden for televerdenen, havde nogle få standarder for grænsefladen til transportnettet kaldet X.21 og X.25.

Der var kun få danske firmaer, der efterfølgende kom med tilbud, og det var (ikke overraskende) Regnecentralen, der fik kontrakten. Trods rygter om det modsatte var der ingen kommunikation mellem de to konsulenter og dem, der skrev tilbuddet, og det var formodentlig Regnecentralens unikke ekspertise på netområdet, der afgjorde sagen. Dermed var udviklingen af datakommunikation i den danske universitetsverden lagt i faste rammer.

Det fysiske net

Centernettets fysiske transportnet var baseret på X.25, den eneste relevante standard for datakommunikation på det tidspunkt. Regnecentralens løsning var at basere transportnettet på et packet-switched net bestående af routere forbundet med fast opkoblede linjer lejet hos teleselskaberne. Jeg var som medlem af Regnecentralens fempersoners Udviklingsråd og som teknisk produktkoordinator med til at skabe en unik computer, RC3502, og dens tilhørende programmeringssprog Real-Time Pascal. Computeren var designet til at køre programmer skrevet i Pascal. Hvor sproget C er udviklet til at lave effektive programmer til fx PDP-computere, vendte vi dette om og designede en computer, der var effektiv til at køre Pascal-programmer; dens instruktions-sæt ligner vore dages JVM (Java Virtual Machine).

Transportnettet blev lavet i et usædvanligt samarbejde mellem de regionale centre, Regnecentralen og teleselskaberne. Resultatet af dette var PAXNET, som i en del år var teleselskabernes interne transportnet. Da RC3502 blev for gammel, erstattede man den med transputer-baserede routere og genbrugte alt det udviklede programmel, og da X.25 blot var et interface til det underliggende pakke-baserede transportsystem, kunne man tilføje IP-teknologi uden større besvær, da dette blev aktuelt.

Applikationssiden

Det var mere besværligt at finde standarder (se faktaboks) for resten af nettet, idet fx TCP/IP ikke var helt færdigt på dette tidspunkt og i øvrigt var en intern amerikansk sag. Man valgte applikationerne fra den britiske 'coloured book'-serie og en transporttjeneste foreslået af IFIP (International Federation for Information Processing).

Der var også store problemer med at få de to centrale services (terminaler og filoverførsel) implementeret på de store mainframecomputere. Dette skyldtes, dels at operativsystemerne var leverandørernes domæne, dels at der ville blive tale om at teste nyt programmel på driftsmaskiner, hvilket generelt er en dårlig ide. Jeg har selv været med til at få en stor ICL-computer hos ESA i Holland til at gå ned under en test, og det var ikke populært, selv ikke, da jeg kunne konstatere, at det skyldtes manglende opdateringer af operativsystemet og ikke en fejl fra min side.

Efter min ansættelse ved DIKU i 1980 fulgte jeg med i arbejdet med Centernettet og var en af de tre første testbrugere via en ledning fra DIKUs maskinstue i Sigurdsgade til RECKUs maskinstue i Vermundsgade. Mit første login dialog-forsøg afslørede, at nettet kørte i snegletempo, og projektmedarbejderne blev lidt bestyrte, da jeg kunne påvise, at det skyldtes et enormt overhead i de valgte protokoller og ikke ineffektiv programmering eller lignende.

Centernettets skæbne

Centernettet mistede gradvis sin betydning efter projektets afslutning i 1983. Et ubekræftet rygte vil vide, at projektlederen bl.a. fik forbud mod at markedsføre nettet. Dette skyldes formodentlig,



Figur 2: Danske netfolk på COST 11 EUTECO-konference i Varese i 1983. Man skal åbenbart tage til Norditalien for at træffe andre danske kolleger. Her har vi taget fri fra en foredragseftermiddag for at snakke. Bagerst i billedet (fra venstre) Alette Brüel (NEUCC), Jørgen A. Richter (NEUCC), den anden projektleder for Centernettet, og Ole Brun Madsen (RECAU), den første projektleder. Forrest Henrik Wendelboe (RECAU), Henrik Heikel Vinther (NEUCC), Peter Reilly (RECAU, gæst fra Trinity College i Dublin) og Erling Knudsen (RECAU). (Privat foto Klaus Hansen)

at de tre centerdirektører med hensyn til transportnettet foretrak andre løsninger. Der er altid en økonomisk afvejning mellem at drive og videreudvikle et transportnet og at købe transportsystemet hos en leverandør, og direktørerne foretrak klart den sidste løsning.

Hvor Centernettet teknisk set måske ikke blev rigtig vellykket, kan projektets rolle som fokus for de danske netaktiviteter ikke overvurderes. Det var projektmedarbejdere, der etablerede det nordiske samarbejde, deltog i udviklingen af danske og internationale standarder og senere formidlede kontakten mellem universitetsverdenen og teleselskaberne. Som udvikler og forsker finder jeg desuden, at det var et meget interessant miljø med mange særdeles kompetente og interessante personer.

DIKU får e-mail i 1983

Jeg var blevet ekstern underviser på datanetkurset på DIKU i januar 1979 og blev fuldtidsansat som adjunkt i maj 1980. På det tidspunkt var datanet og distribuerede systemer et relativt ukendt forskningsområde inden for datalogi, og der var ikke ret meget undervisningsmateriale og kun få kompetente undervisere i universitetsverdenen – specialisterne sad i virksomhederne og udviklede systemer, men fortalte ikke om det.

Der havde i perioden november 1971 til april 1974 været gennemført et større vellykket DIKU-studenterprojekt, NOOS, med det formål at koble RC4000-computeren på H.C. Ørsted Institutet på de tre regionale centre. Herefter havde der ikke været den store aktivitet inden for datanet, før

jeg blev ansat, og en studerende ved navn Keld Simonsen samtidig begyndte at installere UNIX på den ene af DIKUs to PDP 11-computere.

Den første tid på DIKU arbejdede jeg mest med systemprogrammering og objekt-orienteret design anvendt på billedbehandling. Da Kelds aktiviteter førte til, at DIKU fik e-mail til udlandet via Amsterdam i januar 1983, brugte jeg ret hurtigt derefter systemet til at kommunikere med en medforfatter i Californien. Hvor et brev var 5-10 dage undervejs hver vej, klarede e-mail-systemet det på under 6 timer; denne relativt store forsinkelse skyldtes oven i købet, at post først blev ekspederet, når maskinerne kaldte hinanden op, og dette skete ca. hver 4. time. Det blev hurtigt klart, at dette var et værdifuldt arbejdsredskab, og at alle danske forskere burde have adgang til e-mail.

Starten på det internationale arbejde

I 1981 var der større netprojekter i gang i universitetsverdenen både i Norge og Sverige, og det var derfor naturligt, at de nordiske landes projekter skulle mødes for at udveksle erfaringer. Det første nordiske møde afholdtes i Tällberg ved Siljan-søen i Sverige, og her blev det hurtigt klart, at de nordiske landes projekter gik i hver sin retning og ville resultere i inkompatible net. De finske repræsentanter var ikke til megen hjælp, idet de hovedsageligt kun kunne bidrage med beklagelser over, at deres regering ikke var interesseret i netteknologi. Dette ændrede sig radikalt i løbet af et par år, således at Finland blev en meget aktiv partner i såvel det nordiske som det europæiske samarbejde.

Resultatet af mødet blev, at den danske vision om åbne systemer med standardprotokoller blev accepteret som vejen frem, og at Norge stoppede videreudviklingen af egne protokoller. Vigtigt var det også, at man enedes om at søge om nordiske midler til et fælles projekt (Nordunet), herunder at etablere arbejdsgrupper, der skulle se på de tekniske og driftsmæssige detaljer med henblik på at sikre interoperabilitet. Der blev dermed indledt et samarbejde, som stadig foregår, og som har givet flotte resultater. I årene derefter blev det gnidningsløse nordiske samarbejde et forbillede for det europæiske samarbejde.

Samarbejdet havde også sine let komiske elementer. I starten af 1980'erne fremkom lokalnet som Ethernet. På en af de tidlige Nordunet-konferencer var en eftermiddag blevet afsat til møder i arbejdsgrupper, hvoraf en havde overskriften 'lokalnet'. Der dukkede 15-20 personer op, herunder et par universitetsprofessorer, men det viste sig, at alle var kommet for at høre om dette nye, spændende emne - ingen havde praktiske erfaringer at bidrage med. Det blev et kort møde.

Jeg deltog fra starten af i det nordiske samarbejde omkring e-mail (X.400) sammen med medlemmer af Centernet-projektgruppen og holdt jævnlige indlæg på de årlige konferencer. I 1985 organiserede jeg sammen med RECKU Nordunet'85-konferencen i Ebeltoft. Efter Centernetets formelle afslutning fortsatte arbejdet på de regionale centre med de samme medarbejdere, og den internationale koordination blev fortsat set som en væsentlig aktivitet. Oprettelsen af UNI-C ved fusion af de tre centre førte heller ikke til ændringer i de internationale aktiviteter.

Et europæisk samarbejde mellem akademiske net blev etableret ved et møde i Amsterdam i 1986 i form af organisationen RARE (Réseaux Associés pour la Recherche Européenne), og det var naturligt, at deltagerne i de nordiske landes samarbejde også påtog sig at deltage i det europæiske samarbejde.

DIKU-projekter

På de datalogiske institutter og de mindre datacentre ved forskningsinstitutionerne i Danmark var der medarbejdere, der kunne se mulighederne i digital kommunikation mellem computere af moderat størrelse. På DIKU etablerede vi kurser og vejledte projekter med henblik på at uddanne specialister, men manglede udstyr til praktiske eksperimenter. Der blev derfor skrevet et par ansøgninger om udstyr, der ved at give mulighed for tilkobling til Centernettet kunne bruges til projektarbejde mv., men uden held.

De regionale centre havde fokus på at give deres brugerkreds adgang til deres centrale computere og havde dermed ikke nogen udpræget interesse i kommunikation mellem mindre computere på forskningscentre og institutter og dermed heller ikke i at fremme generel datakommunikation. Det havde universitetsverdenen derimod. DIKU og Robin Sharp fra Institut for Datateknik på DTH (i dag DTU) blev derfor penneførere for de ovennævnte institutter og små centre i en ansøgning, som resulterede i en bevilling fra FTU¹ og dermed starten på et net kaldet DUNET (lidt ubeskedent Det danske Universitetsnet - nettet var ret lille). Vi havde dermed dels fået en identitet, dels fået mulighed for at sende e-mail mellem institutionerne (trods det, at ikke alle kørte UNIX) og til og fra udlandet.

Keld Simonsen havde samtidig arbejdet videre med sit projekt og herunder stiftet det navnkundige DKNET med forbindelse til EUNET og dermed UNIX-brugernes internationale net. Da en af DIKUs computere var en vigtig knude i DKNET, havde vi to af nettene i samme hus og dermed gode muligheder for at forbinde dem.

”

*Det var i
høj grad de
inkompatible
protokoller,
der var
problemet*

Projektdeltagere i Danmark

FTU-projektet AUD (Ålborg Universitetsdatacenter)

AUC (Institut for Elektroniske Systemer, Ålborg Universitetscenter)

DAIMI (Datalogisk Institut, Aarhus Universitet)

DIKU (Datalogisk Institut, Københavns Universitet)

DTH/ID (Institut for Datateknik ved DTH)

NEUCC

Århus og Odense Teknika

Forsøgscenter Risø

ISI-DK

JTAS

UNI-C

DIKU og DKNET

Telecom, Datacom og danNet

DDE, DANOSI, Kejo Datacom Aps., Native Communications, Bruel og Rune A/S, Fischer og Lorenz

Kommunedata, DataCentralen, DOU (Datacenteret ved Odense Universitet)

¹ Forskning og Teknologisk Udvikling, et dansk program knyttet til EU's rammeprogrammer. Der er tale om det første rammeprogram for 1983-87.

Samarbejdets kunst

I netmiljøet er der en tendens til at have en monopolistisk tankegang, fx fordi man var der først, fordi man er stor, eller fordi man har de bevilgende myndigheders øre. Det er klart, at teleselskabernes århundredlange monopol på tele- og datakommunikation gjorde det svært for dem at skifte tankegang, da liberaliseringen af sektoren begyndte. Men både UNI-C's og (til en vis grad) DKNET's idé om, at de havde et slags naturligt monopol på digital kommunikation i forskningsverdenen, og at de andre var amatører, gjorde indimellem samarbejdet særdeles vanskeligt.

To danske fløje

Der var omkring 1985 ret forskellige holdninger til, hvorledes man skulle nå frem til et åbent net for forskerne i Danmark.

Den ene gruppe, som traditionelt havde haft rollen som beslutningstagere mht. udviklingen af universitetssektorens computerfaciliteter og inkluderede UNI-C's ledelse, havde en afventende holdning, som var præget lidt af det såkaldte danske mindreværdskompleks: Udlandet er meget bedre til at føre an og lave tekniske løsninger – vi skal bare følge med udviklingen.

Den anden gruppe, som omfattede forskere fra datalogi og udviklere fra it-virksomheder, havde erfaringer med udvikling/implementering af netkomponenter med forskellige ambitionsniveauer og mente, at Danmark havde et forspring, som bl.a. kunne udnyttes til at give danske computer-virksomheder mulighed for at få et stort eksportmarked på basis af et lille offentligt finansieret hjemmemarked. Situationen svarer lidt til nutidens produktion af vindmøller, altså en optimistisk 'vi kan gøre det lige så godt som dem i udlandet (og måske bedre)'-holdning.

Der blev derfor afholdt flere møder med bl.a. Administrationsdepartementet og Forskningssekretariatet på initiativ af institutter og små datacentre for at gøre embedsmænd og politikere opmærksom på, at der var en unik mulighed for at føre en aktiv politik inden for datakommunikation. Det havde desværre ikke den store virkning.

Et broget billede

De danske net udviklede sig derfor uafhængigt af hinanden. DKNET fortsatte sin udbygning baseret på kommunikation ved telefonforbindelser. DUNET brugte den offentlige X.25-baserede DATAPAK-service, faste linjer og telefonforbindelser. UNI-C havde dels sine forbindelser ud til brugerne og stod desuden for den danske side af EARN, et IBM-sponsoreret europæisk net, og for højenergifysikernes og astronomernes net baseret på faste telelinjer. Også protokolmæssigt var der ingen kompatibilitet, idet DKNET brugte protokoller af UUCP-familien, DUNET den åbne standard X.400, EARN IBM-protokoller, og det sidste net DECNET-protokoller.

Fra at have været nabo til RECKU var DIKU flyttet til Universitetsparken 1, men da nogle kurser stadig brugte UNIVAC-mainframen, havde vi et problem med at få en forbindelse mellem de to adresser. Efter en del overvejelser om at grave et kabel ned eller etablere infrarød kommunikation lykkedes det at overtale den daværende regionale teleleverandør, KTAS, til at lave en direkte kabelforbindelse gennem centralerne; et væsentligt problem var, at Jagtvejen dannede grænse mellem to centraldistrikter, og at kablerne derfor gik den forkerte vej. Med denne forbindelse kunne vi bruge de meget billige 70 kbit/s-modem'er.

DIKU havde dermed relativt gode fysiske forbindelser, så det var i høj grad de inkompatible protokoller, der var problemet. Da UNI-C etablerede DENet i 1988, var det derfor relativt simpelt for DIKU at koble sig på, specielt da det groft taget bestod af Ethernet forbundet med faste telelinjer.

På dette tidspunkt (knap fem år efter at DIKU fik e-mail) var e-mail-systemet i Danmark og internationalt så veletableret, at der var behov for en dansk aftale om domænenavngivning, idet administrationen af danske subdomænenavne var en forudsætning for udveksling af e-mail og for, at nettene senere kunne kobles sammen. Aftalen fra november 1987 løste en række problemer ved at præcisere, at DKNET stod for uddelingen af navne i DK-domænet (fx diku.dk, nbi.dk), at nettene forpligtede sig til at videreeksperedere e-mails som ankom fra udlandet, men havde en modtager i et af de andre net, og at økonomiske mellemværender skulle aftales direkte mellem parterne.

Om hvem repræsenterer så de danske holdninger udadtil?

Da Nordunet startede, var det naturligt at Centernettet var den danske deltager. Situationen var anderledes ved stiftelsen af RARE, idet de to fløje i Danmark ikke syntes om, at den anden part skulle repræsentere Danmark. I længere tid måtte RARE's bestyrelse (CoA = Council of Administration) således tolerere, at Danmark alene af alle deltagende lande havde to repræsentanter: Peter Villemoes og mig. Vi havde i øvrigt et udmærket samarbejde, og Peter havde ikke så meget imod de lidt formelle diskussioner i CoA, som jeg havde. Alene at finde navnet RARE tog mange timer, og jeg var ved at blive lidt utålmodig undervejs over, at man ikke tog fat på noget, der efter min mening var mere relevant.

Forskningsprogrammet EUREKA skulle som et af delp projekterne etablere et europæisk forskningsnet. Jeg tog med som teknisk rådgiver til et møde i Bonn i november 1985, hvor EUREKA til alles lettelse blev overtaget til at lade RARE klare kommunikationssiden med projektet CO-SINE (Co-operation for Open Systems Interconnection in Europe), som Danmark efterfølgende besluttede sig for at deltage i.

De danske teleselskaber

Indtil 1987 blev telefonnettet og de forskellige datanetservices leveret af FKT (Fyns Kommunale Telefonselskab), JTAS (Jydsk Telefon-Aktieselskab, stiftet i 1895), KTAS (Kjøbenhavns Telefonaktieselskab, stiftet i 1882) og Statens Teletjeneste, som var en del af P & T (Post- og Telegrafvæsenet) og tog sig af Sønderjylland, kommunikation på landsplan og internationale forbindelser.

I 1991 samledes disse aktiviteter under Tele Danmark A/S, og i 1996 var integrationen fuldført. I perioden 1991-1996 blev visse dele af tjenesterne varetaget af underselskaberne Telecom og Datacom. I 2000 skiftede Tele Danmark navn til TDC.

Netsamarbejdet i Danmark er sket i denne periode med reorganiseringer, og det kan godt forvirre lidt, at partneren først hedder JTAS, dernæst Datacom og til slut Tele Danmark; der er imidlertid tale om de samme personer.

Også i dette projekt viste uenigheden sig. Der blev to tilslutningsmuligheder for Danmark til COSINE's transportnet IXI. På dette tidspunkt var DENet blevet forbundet med de andre nordiske landes tilsvarende net, og den ene forbindelse skete således ved, at Nordunet etablerede en forbindelse i Stockholm til IXI for alle de nordiske lande. Den anden rent danske, som DUNET benyttede sig af, gik via en IXI-knude i Århus. Nu kan man så spørge, hvordan det gik til, at denne knude eksisterede. Det skyldtes, at IXI benyttede PAXNET-teknologien og derfor havde en forbindelse til det jyske teleselskab. Dette var en stille triumf for teknologioptimisterne.

EU-Kommissionen blander sig

Danskernes kontakter i EU-Kommissionen havde set på udviklingen i Danmark med en vis bekymring og havde overraskende nok adgang til midler, der kunne anvendes i et enkelt land til udvikling af dets infrastruktur – normalt var det et krav fra EU's forskningsprogrammer, at der skulle være tale om samarbejde over landegrænser. Vi fik en henvendelse om, at hvis de danske net kunne enes om at samarbejde, kunne vi få midler til hardware og software, arbejdstimer og rejser.

Dette skete bl.a. med inddragelse af formanden for DANDOK, Kjeld Rasmussen, som samtidig repræsenterede Danmark i COSINE Policy Group.



Figur 3: Samarbejdspartnere i det danske projekt ISI-DK ved en lille reception i anledning af den første EU-kontrakt: forrest Jens Bech Andersen (Jydsk Telefon/TDC Datacom), bagerst tv. Klaus Hansen (DUNET/DIKU) og th. Ole Carsten Pedersen (UNI-C). Keld Simonsen fra DKNET er ikke med, da DKNET var underleverandør til DIKU. Fotoet er taget til en af de meget få pressemeddelelser fra samarbejdet og blev bragt i Berlingske Tidende Erhverv 2. juni 1992. (Foto ISI-DK 1992)

Der skulle holdes et møde om, hvorledes den praktiske side af et samarbejde kunne etableres, men da en del af netmiljøet anså det for problematisk at invitere UNI-C med, blev resultatet, at Kjeld Rasmussen som privatperson indbød en række andre privatpersoner med interesse for net til at komme hjem til ham i Søborg, hvor de så kunne møde en EU-repræsentant, der tilfældigvis var i landet. Alle blev lidt overraskede, da to medarbejdere fra UNI-C ringede på døren, da mødet var ved at starte, og bad om at måtte komme med. Det fik de så lov til, og resultatet blev først et projekt om transportnet og e-mail baseret på X.400 og dernæst endnu et projekt om X.500 finansieret for en stor del fra EU's VALUE II-program, men med intern medfinansiering. Datacom (Tele Danmark), UNI-C og DIKU var partnere i kontrakten med DKNET som underleverandør. I arbejdet deltog også en række danske virksomheder (se faktaboks).

Man kan kalde det utidig indblanding i et medlemslands interne sager, men det klarede luften, og vi fik et fint samarbejde ud af det.

Men alt er dog ikke idyl

På et vist tidspunkt gik datatrafikken mellem DKNET og resten af Danmark via Californien. Nogle af de markante personligheder i netmiljøet havde vanskeligt ved at lave en aftale om at udveksle trafik internt i Danmark, og den tekniske direktør på UNI-C Hans Ole Aagaard og jeg holdt derfor et møde med de pågældende for at finde frem til, hvori problemet lå. Den ene så allerede på dette tidspunkt de sikkerhedsmæssige problemer ved sammenkobling af to net med forskellig sikkerhedspolitik, og UNI-C ville meget nødtigt have forstyrrelser i netdriften forårsaget af trafik til og fra DKNET. Det lykkedes efter en del diskussion at nå frem til en løsning i form af en telelinje og diverse filtre mellem DIKU og Symbion, hvor DKNET holdt til. Dette er en tidlig udgave af DIX = Danish Internet Exchange.

Det 'rigtige' internet kommer til Danmark i 1989

Jeg stod på et vist tidspunkt omkring 1986 på Strandvejen nord for København og kiggede over på Skånes kyst og overvejede, hvorledes man bedst kunne krydse Øresund og få forbindelse til de svenske net. Det egentlige problem var økonomien, idet det var ret dyrt at etablere fysiske net. De første Ethernet-tilslutninger kostede 6.000-7.000 kr. pr. computer, og det var derfor alt for dyrt at koble en enkelt pc på, selvom de første pc'er kostede 30-40.000 kr. Brugte man DATAPAK til forbindelser til andre institutioner, kostede dette i 1993 dels 17.000 kr./år i abonnement for at få 9.600 bit/s og 75 kr./MB i Danmark og 350 kr./MB mellem europæiske lande. Faste linjer mellem danske abonnenter kostede tilsvarende 16.800 kr./år ved nabocentraler og 45.000 kr./år for de længste danske distancer. Faste linjer i Norden og Europa og over Atlanterhavet var endnu dyrere. Der var derfor god fornuft i at dele udgifterne ved at samarbejde.

I 1987 var man kommet i gang med at diskutere idéen om et nordisk multiprotokolnetværk ('åbent' transportnet). På Nordunet '87-konferencen i Espoo ved Helsinki oktober blev dette diskuteret i Nordunets styregruppe. En gæst på konferencen, Larry Landweber fra USA, sad foran mig i bussen, som en aften skulle transportere os til en svensk teaterforestilling, og jeg spurgte ham undervejs, om han havde et godt råd om forbindelser til Internettet i USA. Han svarede "Hvad om I delte en linje?", hvilket blev hørt af de medlemmer af styregruppen, der sad i nærheden, og Larry blev derefter inddraget i arbejdet. Dette viser, at udviklingen sommetider sker ved en række små puf, som udløser en større reaktion.



Udviklingen sker somme-tider... ved en række små puf, som udløser en større reaktion



Resultatet af diskussionerne blev en beslutning om at oprette forbindelser mellem Lyngby, Oslo og Helsinki til datacenteret ved KTH (Kungliga Tekniska Högskolan) i Stockholm og derfra etablere en forbindelse til USA, som man deltes om at finansiere, og dette blev sat i værk i den efterfølgende tid. Forbindelsen Lyngby-Stockholm stødte ind i problemer og blev derfor oprettet sent i 1988 som den sidste.

Afrunding

Med forbindelsen til internettet var det helt klart, at fremtiden lå i en teknik, der hovedsagelig var baseret på TCP/IP og kun undtagelsesvis på 'rigtige' internationale standarder fra ISO og CCITT. Stort set alle leverandører var begyndt at inkludere TCP/IP i deres operativsystemer, og da internettets arkitektur faktisk var åben, var den ønskede fleksibilitet til stede. De forskningsinstitutioner i Danmark, som ikke kunne få en direkte internetopkobling, kunne stadig benytte internettets applikationer ved hjælp af forbindelser baseret på X.25 eller lignende.

I 1983 havde jeg et indlæg på Nordunet '83-konferencen i Helsinki med titlen "The Next Seven Years". I indlægget fremlagde jeg en række krav til netværk for forskere og en liste over problemer, som skulle løses. Det er interessant at se tilbage på et oplæg med så ambitiøs en titel - at se syv år ud i fremtiden.

Kravene er på skrivende tidspunkt stort set alle opfyldte. En undtagelse er ønsket om en arkiverings-service for sikker lagring af forskningsdata i lang tid. Idéen om en computer på ens skrivebord, hvorfra man har netadgang til hele verden, herunder e-mail og filoverførsel, er blevet en realitet selv i private hjem.

Denne beretning slutter, hvor internettet og world wide web begynder. Det var i 1983 ikke muligt at forudse de mange måder, en så enkel protokol som HTTP (HyperText Transfer Protocol) kunne bruges på, og heller ikke omfanget af den information, der efterhånden ville blive tilgængelig på internettet. Det var også vanskeligt at forudse, at kombinationen af ISO's langsomme arbejdstempo og høje ambitionsniveau og de amerikanske internet-pionerers høje tempo og mere beskedne ambitionsniveau ville afgøre, hvilken teknologi der kom til at dominere.

I 1995 var de danske projekter ved at være afsluttede, og de kommercielle serviceudbydere spillede en stadig større rolle. Da forskningsmiljøet omkring de generelle net endvidere stort set var forsvundet, skiftede jeg forskningsområde, idet jeg opgav datanet og genoptog billedbehandling. Dermed var en interessant periode slut, og jeg har derefter blot været almindelig bruger af internettet. ♦

Læs mere

Hans Ole Aagaard. Jubilæum 1965-2005: På forkant i 40 år. UNI-C, oktober 2005.

Martin Bech. History of DENet (<http://www.denet.dk/history/>)

Klaus Hansen. The next seven years - a university user's view. In Conference Proceedings NORDUnet 83, Helsinki, October 1983.

Kaarina Lehtisalo. The History of NORDUnet. Nordunet, May 2005.

Forkortelser og begreber

COSINE	Cooperation for Open Systems Interconnection in Europe, et EUREKA-projekt.
COST 11	EU-program for EU's medlemmer plus en række lande, herunder de nordiske lande. RARE, COSINE mv. omfattede COST 11-landene.
DANDOK	Dansk organisation med formålet at give forskere adgang til referencedatabaser verden over.
DECnet	Digital Equipment Corporations protokolfamilie brugt i VAX-computere, der brugte operativsystemet VMS, der var meget populært hos fx fysikere i hele verden.
DENet	Danish Educational Net, 1987-1998, forgængeren for Forskningsnettet.
DKNET	Danske UNIX-brugeres net i perioden fra 1982 til 1996, hvor Tele Danmark købte det.
DUNET	Dansk net hovedsageligt beregnet til e-mail.
EARN	European Academic Research Network baseret på IBM-produkter og sponsoreret af IBM, 1984-87.
Ethernet	Lokalnet specificeret i IEEE 802.
EUnet	European Unix-user's network.
EUREKA	EU-forskningsprogram.
FTU	Dansk forskningsprogram.
ISI-DK	Navn for det danske netsamarbejde i forbindelse med VALUE-projekterne.
IXI	Europæisk transportnet etableret i forbindelse med COSINE.
NETSAM	Koordineringsgruppe for de danske akademiske net, især for Københavnsområdet.
NEUCC	Se faktaboks om edb-centre.
Nordunet	Løst samarbejde i Norden, som bl.a. afholdt en årlig konference.
NORDUnet	Nordisk datanet startet 1989.
NORDUNET	Nordisk udviklingsprogram, 1985-1991.
RARE	Réseaux Associés pour la Recherche Européenne.
RC79	Regnecentralen af 1979; se Regnecentralen.
RECAU	Se faktaboks om edb-centre.
RECKU	Se faktaboks om edb-centre.
Regnecentralen	Regnecentralen (RC) blev oprettet i 1955 under Akademiet for de Tekniske Videnskaber (ATV) og udviklede i 1956 den første danske computer, DASK, og i 1958-61 efterfølgeren GIER og er dermed den første danske it-virksomhed. I 1964 blev RC omdannet til aktieselskab, som bl.a. udviklede og producerede computere og software. Efter betalingsstandsningen i 1979 fortsatte firmaet nogle år, men blev opslugt af det britiske firma ICL og er nu en del af Fujitsu-Siemens.
TERENA	Fusion af EARN og RARE.
UNI-C	Se faktaboks om edb-centre.
VALUE	EU-program, der bl.a. havde midler til at støtte infrastrukturprojekter.
X.25	CCITT-standard for datakommunikation, i Norden realiseret ved DATAPAK-tjenesten.
X.400	International standard for håndtering af e-mail.
X.500	International standard for opslagstjenester og forbilledet for Active Directory og LDAP.

KLAUS HANSEN



Klaus Hansen er kandidat i datalogi og matematik fra DIKU i 1975 og har i perioden 1973-1980 arbejdet på Regnecentralen som systemprogrammør, -udvikler og -arkitekt. Efter årelang tilknytning til DIKU, bl.a. som GIER-operatør, instruktør og undervisningsassistent, blev Klaus adjunkt på DIKU i 1980 og

lektor i perioden 1984-2009. Klaus har bl.a. som studerende siddet i DIKUs første bestyrelse og været redaktør af DIKUs første kandidatstudieordning i 1971.

Klaus har derudover siddet i Dansk Standardiseringsråd for DIKU i 25 år og i fem år været medlem af Open Source OCES-koordinationsgruppen i Videnskabsministeriet. Han har desuden været medlem af Videnskabsministeriets it-sikkerhedspanel. På publikationslisten står bl.a. "It-lex", hvor han var redaktør og bidragsyder. I dag er Klaus ekstern lektor på DIKU og ved EBUSS-uddannelsen på IT-Universitetet i København samt formand for Informationssikkerhedsudvalget ved Det Naturvidenskabelige Fakultet.



Nogle programmeringssprog er mere lige end andre

.....
Af **Torben Mogensen**, DIKU
.....

Der findes tusindvis af forskellige programmeringssprog, og der bliver dagligt designet nye. Men er der virkelig så stor forskel på forskellige sprog, og hvis ikke, hvorfor så i det hele taget lave nye sprog?

Begyndelsen: Church og Turing

Allerede inden der fandtes egentlige computere, var matematikere og logikere i gang med at definere modeller for automatiserede beregninger. De mest kendte af disse var englænderen Alan Turing (1912-1954) og amerikaneren Alonzo Church (1903-1995). De havde i 1930'erne udviklet hver deres model for automatiske beregninger: Turings model er kendt som turingmaskinen, og Churchs model som lambdakalkulen. Se tekstboksene på de følgende sider for detaljer om de to modeller.

Hverken turingmaskinen eller lambdakalkulen var tænkt som forslag til fysiske beregningsmaskiner. Turingmaskiner kunne godt realiseres med datidens teknologi (hvis man ser bort fra strimlens uendelighed), men de var for bøvlede at bruge: Selv forholdsvis enkle beregninger såsom at lægge to tal sammen krævede ret komplicerede turingmaskiner. Lambdakalkulen var noget nemmere at bruge, men der var ikke nogen oplagt måde at lave en maskine, der kunne foretage beregninger i lambdakalkule.

Både turingmaskinen og lambdakalkulen var primært designet som værktøjer til at filosofere over hvilke problemer, der i det hele taget kan lade sig gøre at løse med automatiske beregninger. Derfor var fokus i designet på at gøre det nemt at lave matematiske beviser for deres egenskaber uden det store hensyn til realiserbarhed eller brugbarhed til andre formål. Det blev for eksempel bevist, at det er muligt at definere matematiske funktioner, der ikke kan beregnes i disse to modeller for beregning.

Når man har to så forskellige modeller for automatiske beregninger som turingmaskinen og lambdakalkulen, er det selvfølgelig interessant at vide, om de kan det samme, eller om den ene kan lave beregninger, som den anden ikke kan. Sammen med logikerne Stephen Kleene og J.B. Rosser beviste Church og Turing, at enhver beregning, der kan laves med en turingmaskine, også kan laves i lambdakalkulen og omvendt. Så hvis man skal filosofere over grænserne for automatiske beregninger, er det ét fedt, om man bruger turingmaskiner eller lambdakalkulen i sine tankeeksperimenter. Men hvad med andre modeller? Kunne det tænkes, at en helt tredje model for beregninger kunne mere end både turingmaskiner og lambdakalkule?



Alan Turing



Alonzo Church

Det mente Church og Turing ikke og formulerede derfor en hypotese, som i dag er kendt som Church-Turing-tesen: Enhver beregning, der kan laves som en mekanisk proces, kan også laves af en turingmaskine eller i lambdakalkulen. Derfor ville et bevis for, at en bestemt funktion ikke kan beregnes med en turingmaskine (eller i lambdakalkulen), med det samme bevise, at ingen mekanisk proces kan lave denne beregning.

Church-Turing-tesen kan ikke bevises matematisk, da der ikke er nogen matematisk definition af, hvad en mekanisk proces er. Men tesen er generelt accepteret som sand, da ingen endnu har kunnet beskrive en mekanisk proces for beregning, som ikke kan eftergøres med en turingmaskine eller i lambdakalkulen.

Et programmeringssprog såsom Java eller C er også en model for automatiske beregninger: Man beskriver en beregning som et program i sproget og kan efterfølgende lave beregningen på en computer. Så det er naturligt at spørge, om et givent programmeringssprog kan lave alle beregninger, der kan laves af en turingmaskine eller lambdakalkulen. I givet fald kaldes sproget for turingkomplet. Langt de fleste gængse programmeringssprog (for eksempel Java og BASIC) er turingkomplette og kan derfor bruges til at beskrive alle tænkelige automatiske beregninger. Så set ud fra et matematisk synspunkt kan det være hip som hap, hvilket programmeringssprog, man bruger til at programmere sin computer. Så hvorfor i det hele taget lave nye programmeringssprog?

”

Langt de fleste gængse programmeringssprog (for eksempel Java og BASIC) er turingkomplette

Turingmaskinen

En turingmaskine består af fire komponenter:

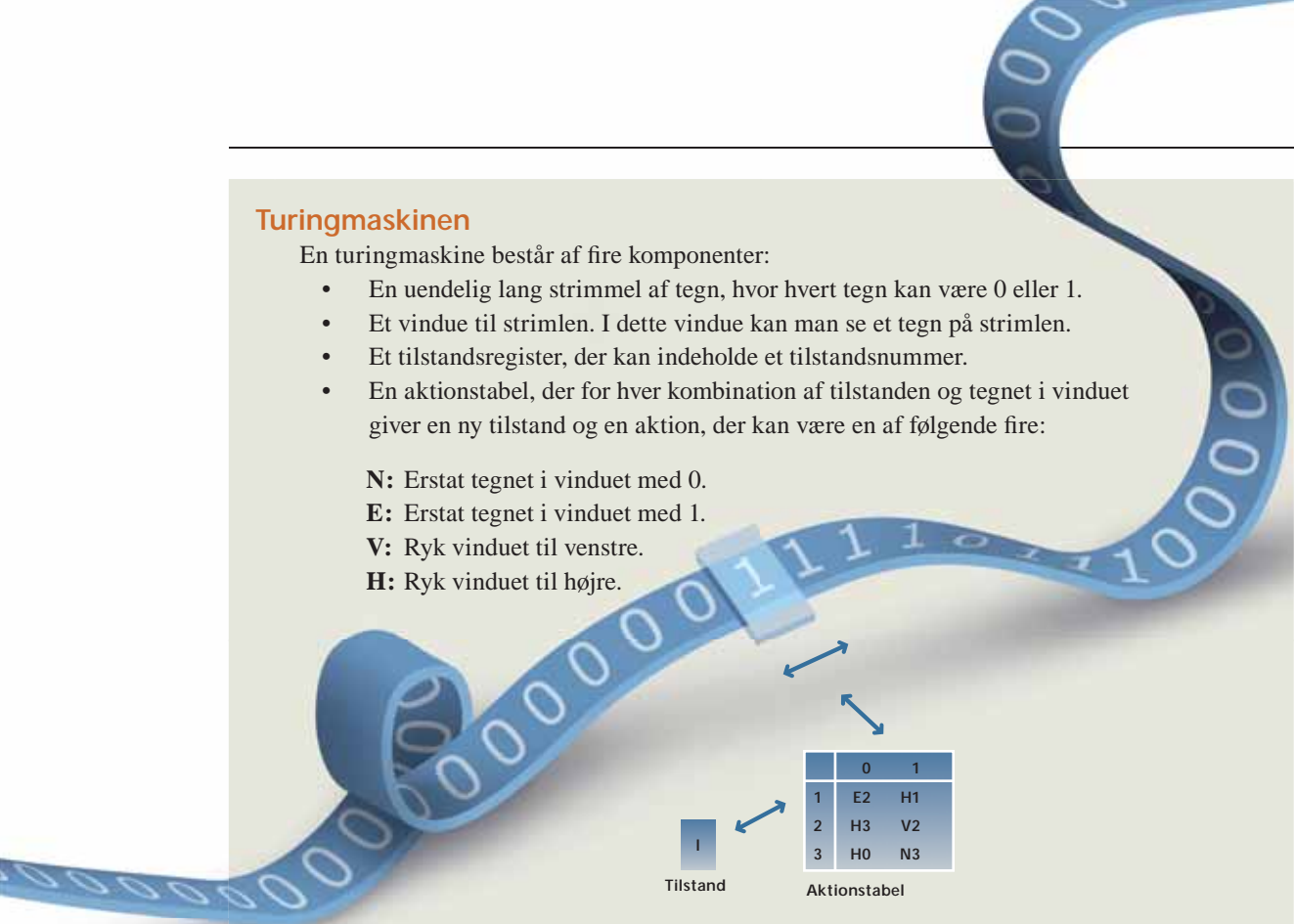
- En uendelig lang strimmel af tegn, hvor hvert tegn kan være 0 eller 1.
- Et vindue til strimlen. I dette vindue kan man se et tegn på strimlen.
- Et tilstandsregister, der kan indeholde et tilstandsnummer.
- En aktionstabel, der for hver kombination af tilstanden og tegnet i vinduet giver en ny tilstand og en aktion, der kan være en af følgende fire:

N: Erstat tegnet i vinduet med 0.

E: Erstat tegnet i vinduet med 1.

V: Ryk vinduet til venstre.

H: Ryk vinduet til højre.



	0	1
1	E2	H1
2	H3	V2
3	H0	N3

Tilstand Aktionstabel

Man kan betragte strimlen som lageret i en computer og aktionstabellen som programmet. Ved starten af beregningen er alle tegn på strimlen nuller undtagen under og til højre for vinduet, hvor inddata til beregningen er repræsenteret som et endeligt antal nuller og ettaller, hvorefter strimlen igen kun indeholder nuller. Tilstandsregistret indeholder ved start tilstandsnummer 1, og man sætter derefter maskinen i gang med at lave skridt. Hvert skridt består i at gøre det, som aktionstabellen angiver. Når tilstandsregistret bliver 0, stopper maskinen, og uddata af beregningen ligger under og til højre for vinduet efterfulgt af nuller.

Man kan repræsentere et tal n som n 1'ere efterfulgt af et 0, så en turingmaskine, der beregner summen af to tal, vil starte med en strimmel, hvor de to inddatatall m og n er repræsenteret som m 1'ere, et 0, n 1'ere og derefter 0'er på resten af strimlen. Ved afslutning vil strimlen indholde $m+n$ 1'ere og ellers 0'er. Det kan man gøre med den aktionstabel, der er vist i billedet, hvor man slår op med den nuværende tilstand (1, 2 eller 3) og tegnet i vinduet (0 eller 1) og finder en aktion (N, E, V eller H) og en ny tilstand (0, 1, 2 eller 3).

Først rykkes vinduet til højre indtil det 0, der adskiller de to tal. Det erstattes af et 1, og vinduet rykkes til venstre indtil 0'et før det første tal. Derefter rykkes vinduet en gang til højre, og 1'et udskiftes med et 0, hvorefter der rykkes endnu en gang til højre, og maskinen stopper.

Lambdakalkulen

Lambdakalkulen har navn efter det græske bogstav λ , som udtales “lambda”. Lambdakalkulen består af symbolske udtryk og regler for, hvordan man kan reducere udtryk til værdier – ligesom regneregler beskriver, hvordan et regneudtryk kan reduceres til et tal.

- Et udtryk kan være en variabel. Vi bruger små bogstaver som fx. a , b , x og y til variable.
- Et udtryk kan have formen $(M N)$, hvor M og N også er udtryk.
- Et udtryk kan have formen $\lambda x.M$, hvor x er en variabel, og M er et udtryk, der kan indeholde variablen x .

Det er endvidere et krav, at forskellige forekomster af λ skal efterfølges af forskellige variable. Fx. er udtrykket $\lambda x.\lambda x.x$ ikke et lovligt udtryk, da de to forekomster af λ begge efterfølges af variablen x .

Et udtryk af formen $\lambda x.M$ siges at være ækvivalent med et udtryk af formen $\lambda y.N$, hvis M og N er ens, bortset fra at M bruger x alle de steder, hvor N bruger y , og omvendt. For eksempel er udtrykkene $\lambda x.(a x)$ og $\lambda y.(a y)$ ækvivalente.

Der er kun én reduktionsregel i Lambdakalkule: Et deludtryk af formen $(\lambda x.M N)$ reduceres til et kopi af udtrykket M , hvori alle forekomster af x er erstattet med udtryk, der er *ækvivalente med* N . Hvis man uden videre erstatter alle forekomster af x med N , kan man overtræde forbuddet mod gentagne variable efter λ , men ved at bruge ækvivalente udtryk kan man undgå dette. For eksempel reduceres $(\lambda x.(x x) \lambda y.y)$ til $(\lambda a.a \lambda b.b)$, idet de to forekomster af x er erstattet med de til $\lambda y.y$ ækvivalente udtryk $\lambda a.a$ og $\lambda b.b$.

En normalform er et udtryk, der ikke kan reduceres med denne reduktionsregel. Programmer er vilkårlige udtryk, og værdier er udtryk i normalform. Man kører et program P med inddata D ved at reducere udtrykket $(P D)$ til normalform. Man kan repræsentere naturlige tal som værdier i Lambdakalkule på følgende måde: 0 repræsenteres som udtrykket $\lambda x.\lambda y.y$, 1 som $\lambda x.\lambda y.(x y)$, 2 som $\lambda x.\lambda y.(x (x y))$, 3 som $\lambda x.\lambda y.(x (x (x y)))$ og så videre.

Vi kan definere et program P , som beregner N^N , hvor N er et naturligt tal, som $P = \lambda n.(n n)$. Vi kan for eksempel køre P med repræsentationen af tallet 2 som inddata:

```
(λn.(n n) λx.λy.(x (x y)))
→ (λa.λb.(a (a b)) λc.λd.(c (c d)))
→ λb.(λe.λf.(e (e f)) (λg.λh.(g (g h)) b))
→ λb.(λe.λf.(e (e f)) λh.(b (b h)))
→ λb. λf.(λi.(b (b i)) (λj.(b (b j)) f))
→ λb. λf.(λi.(b (b i)) (b (b f)))
→ λb. λf.(b (b (b (b f))))
```

hvor pile angiver reduktioner. Bemærk, at vi flere gange omdøber udtryk for at undgå forbuddet mod gentagelse af navne efter λ . Det resulterende udtryk er, som forventet, ækvivalent med repræsentationen af 4.

Hvornår er et sprog bedre end et andet?

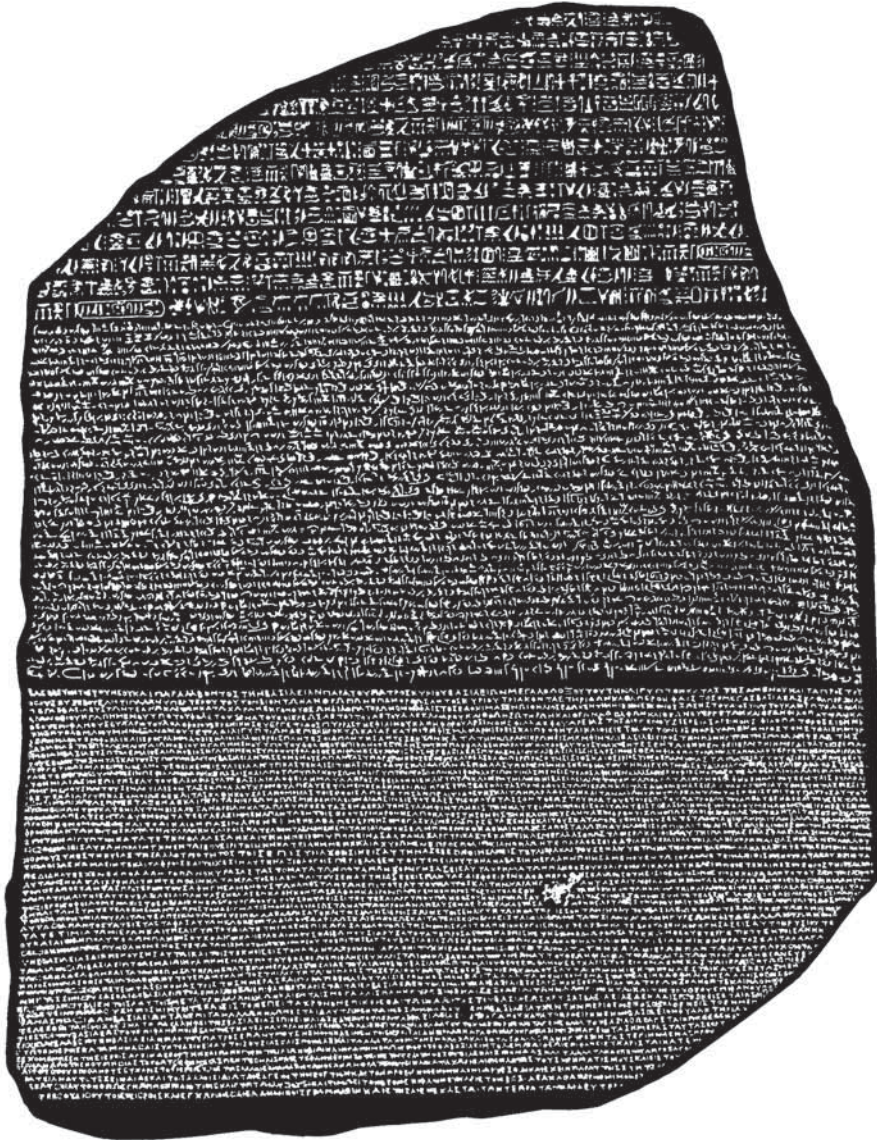
Når to programmeringssprog begge er turingkomplette, er de som sagt på en måde lige gode: De kan beregne de samme ting. Hvis man vil diskutere, om et programmeringssprog er bedre end et andet, må man derfor i de fleste tilfælde bruge andre kriterier end hvilke beregninger, det er muligt at lave i sprogene.

Der har været mange forskellige forslag til sammenligningskriterier for programmeringssprog. Her er nogle af de mest almindeligt brugte kriterier:



Også runer findes i skandinaviske varianter.

Rosettastenen, som stammer tilbage fra de gamle Egypten, viser, at man allerede i oldtiden anvendte forskellige sprog til at formidle det samme budskab.



Programmers størrelse:

Hvis et program i et sprog A altid kan omskrives til et kortere program i sproget B, er B et bedre sprog.

Hastighed:

Hvis et program i et sprog A altid kan omskrives til et hurtigere program i sproget B, er B et bedre sprog.

Nemhed:

Hvis det for alle beregninger er nemmere at lave et program til beregningen i sproget B, end det er i sproget A, så er B et bedre sprog.

Men der er også problemer med alle disse kriterier:

Størrelse:

Man kan bevise, at hvis både A og B er turingkomplette sprog, så kan alle tilstrækkelig store programmer i sproget A omskrives til programmer i sproget B, som højst er en smule større. Med andre ord er det kun for små programmer, at der kan være en væsentlig forskel på, hvor meget tilsvarende programmer fylder i de to sprog. Endvidere kan det være umuligt at afgøre, om et givent program er det mindst mulige program til at lave en given beregning i et givent sprog.

Hastighed:

Et programmeringssprog kan implementeres på forskellige computere på mange forskellige måder, så hastigheden af programmer i sproget er afhængig dels af implementeringsmetoden for sproget og dels af den computer, sproget er implementeret på. Man kan derfor ikke give noget almenlydigt mål for, hvor hurtigt et program er. Man kan allerhøjst sige, at på denne specifikke maskine med denne specifikke implementering af de to sprog, kører dette specifikke program i sproget A hurtigere end dette specifikke tilsvarende program i sproget B. Og selv når man gør tidsforbruget veldefineret ved at fastlægge sprogets implementering og maskine, er det, ligesom det er umuligt at bevise, at et program er det korteste til at lave en bestemt beregning, også umuligt at vise, at et specifikt program er det absolut hurtigste til en bestemt beregning.

Nemhed:

Hvor nemt det er at skrive et program, er en meget subjektiv sag: Én programmør kan synes, at det er nemmere at skrive programmer i sproget A, mens en anden kan synes, at det er nemmere at skrive i sproget B. Så man kan ikke lave et objektivt mål for nemhed. Man kan lave studier af udvalgte menneskers erfaring med forskellige programmeringssprog, men resultatet af et sådant studie vil altid afhænge af de udvalgte mennesker – deres uddannelse, kultur og personlighed.

Der vil derfor altid være masser af forbehold, når man skal sammenligne programmeringssprog. Ikke desto mindre finder man mange diskussioner, hvor folk prøver at overbevise andre om deres personlige favoritsprogs overlegenhed uden at lave de nødvendige forbehold. Man ser fx. ofte udsagn som “C er et hurtigere sprog end Java”, “Det er nemmere at skrive programmer i Javascript end i C”, “Programmer i Perl er kortere end programmer i Java” og så videre. Men den slags påstande kan man ikke bruge til noget – de siger mere om de personer, der laver udsagnene, end om de sprog, udsagnene omhandler.

Det betyder dog ikke, at man overhovedet ikke kan sammenligne turingkomplette programmeringssprog: Hvis man laver passende forbehold, kan man godt sige, at et sprog under disse specifikke forbehold opfylder et bestemt kriterium bedre end et andet. Man kan for eksempel vælge en bestemt computertype og de mest udbredte implementeringer af udvalgte programmeringssprog til denne computertype og sammenligne programmer, der er skrevet af erfarne programmører med henblik på at løse specifikke problemer enten med det kortest mulige program eller i kortest tid. Så kan man godt sige, at i denne kunstige situation var programmerne i det ene sprog typisk kortere eller hurtigere end programmerne i det andet sprog.

Hvis man gør lignende forbehold klart, kan man også forsøge at designe et sprog, der udmærker sig på en særlig måde under disse specifikke forbehold. Men det er meningsløst at sammenligne eller designe programmeringssprog uden at have gjort sig klart, hvilke forbehold man har taget, og hvilke kriterier sammenligningen eller designet laves under.



Et svagt sprog kan have fordele, som turingkomplette sprog ikke har

Svage sprog

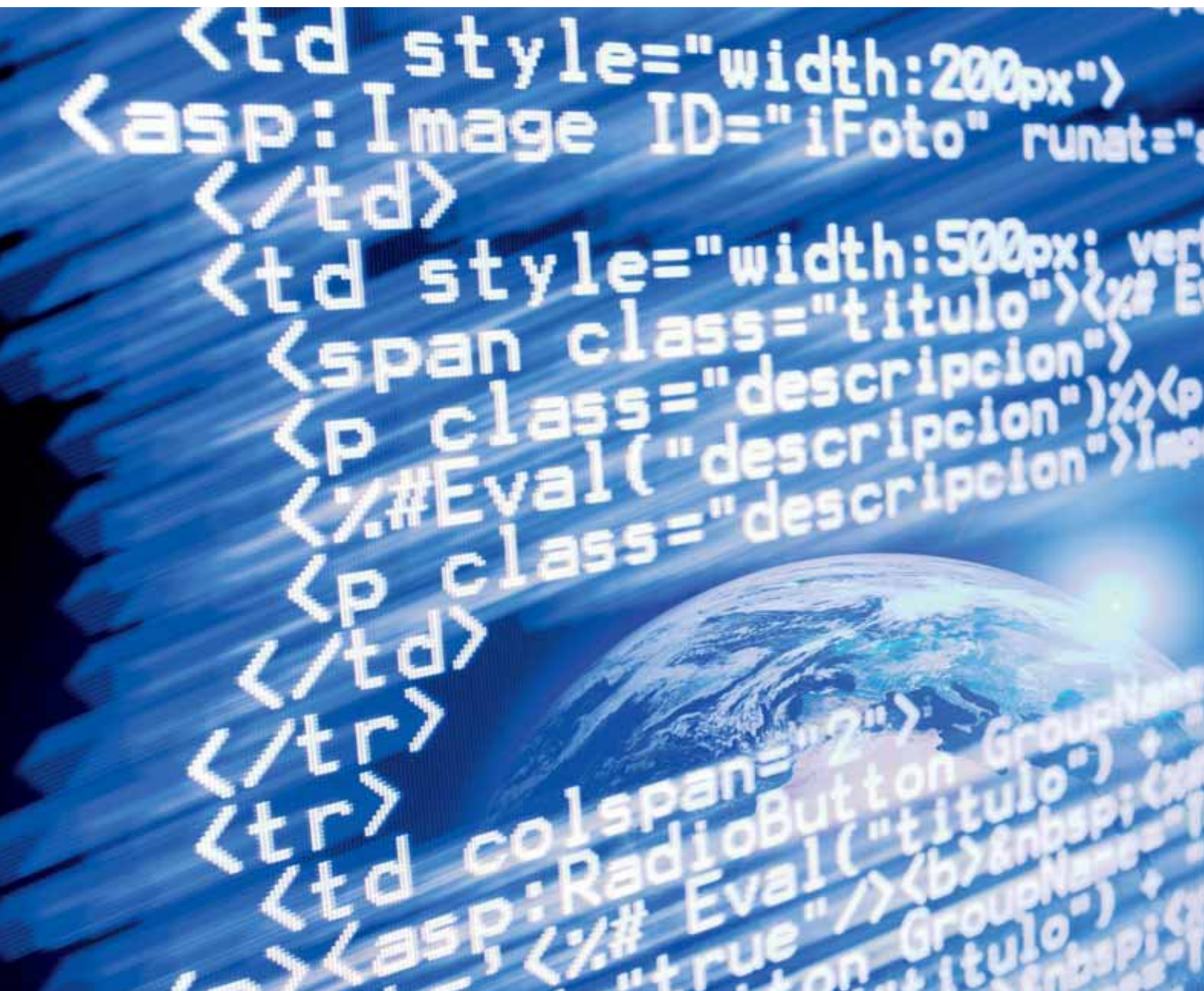
Et sprog, der ikke er turingkomplet, kan pr. definition ikke lave alle tænkelige automatiske beregninger, og man kan derfor kalde det et svagt sprog. Umiddelbart kunne man tro, at det er meningsløst at bruge svage programmeringssprog, når der findes så mange stærke (turingkomplette) sprog. Men et svagt sprog kan have fordele, som turingkomplette sprog ikke har:

- Det kan være umuligt at bevise korrekthed af programmer i turingkomplette sprog, men man kan lave svage sprog, hvor det altid er muligt automatisk at afgøre, om et program overholder en given formel specifikation.
- Det kan være umuligt at give øvre grænser for tids- og pladsforbrug for programmer i stærke sprog, men svage sprog kan garantere, at køretid og plads er begrænsede.
- Det kan være meget nemmere at implementere et svagt sprog effektivt, end det er at implementere et stærkt sprog effektivt. Det kan for eksempel gælde, hvis man vil køre programmer på højparallelle computere. Der kan også være tilfælde, hvor der kendes effektive metoder til at lave beregninger inden for et bestemt problemområde, men hvor disse metoder er meget komplicerede. I denne situation kan man lave et sprog, der kun kan bruges til dette problemområde, men hvor implementeringen af sproget sørger for, at de effektive metoder altid bruges, uden at programmøren behøver at bekymre sig om det.

Derfor er der en stigende tendens til, at der bliver designet stærkt specialiserede svage sprog. Det, man taber i fleksibilitet og beregningsstyrke ved at bruge et svagt sprog, vinder man på andre områder: sikkerhed, garantier for ressourceforbrug, bedre effektivitet og så videre.

Man kan i reglen godt opnå den samme sikkerhed, effektivitet og ressourcegaranti i et turingkomplet sprog, men det vil kræve, at der for hvert enkelt program laves bevis for disse egenskaber, hvor et svagt sprog kan garantere dette for alle programmer, der kan skrives i dette sprog. Man kan groft sagt sige, at et svagt sprog er for dovne programmører: En omhyggelig programmør kan lave et lige så godt eller bedre program i et turingkomplet sprog, men det kræver ofte meget mere arbejde både at skrive programmet og efterfølgende at vise, at programmet opfylder de stillede krav.

Man kan sammenligne svage og stærke sprog med værktøj: Man kan ikke save med en skruetrækker, og man kan ikke skrue skruer i med en sav, men man kan begge dele med en schweizerkniv. Men hvis man skal skrue en skrue i, så er det nemmere at bruge skruetrækkeren end schweizerkniven, og hvis man skal save, er det nemmere at bruge saven end schweizerkniven. Det, man mister i

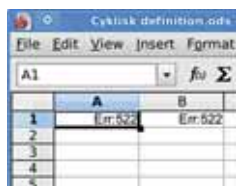


HTML er et eksempel på et svagt sprog, der har fået stor anvendelse.

fleksibilitet med de specialiserede værktøjer, vinder man i brugbarhed til specifikke opgaver. Og hvor sammenligning af turingkomplette sprog kræver en masse forbehold og subjektivitet, kan man ofte lave uforbeholdne sammenligninger mellem svage sprog: Man kan vise, at et sprog A kan lave flere beregninger end et sprog B, eller at et sprog A kan give flere garantier om programmers opførsel end et sprog B. Herunder er nogle eksempler på svage sprog med beskrivelser af de fordele, man opnår ved at sætte begrænsninger i sprogenes beregningsstyrke, og de ting, man mister ved at sætte disse begrænsninger.

Regneark

Regneark såsom Microsofts Excel og Open Office Calc er også programmeringssprog: Man kan beskrive beregninger og få dem udført automatisk. De mest avancerede regneark har efterhånden fået så mange faciliteter, at de er turingkomplette, men i grundformen er de ikke: Hver celle kan indeholde enten et tal, en formel eller en tekst. Formler kan bruge indholdet af andre celler til at beregne en værdi for sin egen celle, men det er ikke tilladt at have cykliske afhængigheder mellem formlerne. For eksempel kan cellen A1 ikke indeholde formelen `=B1+1`, samtidig med at cellen B1 indeholder formelen `=A1+1`, for ingen af disse formler kan beregnes, før man kender værdien af den anden celle. Manglen på cykliske afhængigheder sikrer, at beregningen kan ske hurtigt: Uanset hvor mange celler der ændrer indhold, behøver man kun at genberegne hver formel én gang for at få det færdige resultat – forudsat, at man regner formlerne i den rigtige rækkefølge. Denne rækkefølge kan man også finde i forholdsvis kort tid, så hele genberegningsprocessen er hurtig (forudsat, at de enkelte formler ikke tager lang tid at beregne). Denne garanti for hurtig beregning vil man ikke have i et turingkomplet sprog. Man kan ikke sige, at regneark er specialiserede programmeringssprog: De bliver brugt til mange forskellige slags beregninger lige fra en skoles udflugtsbudget til styrken af en bropille. Fordelene ved regneark kan derfor ikke siges at stamme fra en indskrænkning af problemområdet men snarere en indskrænkning af kompleksiteten af de beregninger, der kan udtrykkes i regnearkene.



Eksempel på, at cyklisk afhængighed er umulig i regneark: Cellen A1 kan ikke indeholde formelen `=B1+1`, samtidig med at cellen B1 indeholder formelen `=A1+1`.

Stærkt specialiserede sprog

En anden måde at opnå fordele gennem svage sprog er at designe dem til at løse opgaver inden for et meget specifikt problemområde: Mange detaljer ved løsningen kan underforstås og behøver derfor ikke at beskrives eksplicit i programmerne, og man kan lave automatiske analyser af programmerne, så man kan få oplysninger om deres egenskaber.

Et eksempel er sproget Troll: Det eneste, man kan programmere i Troll, er forskellige måder at kaste med terninger på og kombinere terningernes værdier til et enkelt tal. For eksempel vil programmet `max 3d10` give resultatet af at kaste 3 tisedede terninger (`3d10`) og finde den største af de tre værdier (`max`). Da terningslag er tilfældige, vil det samme program give forskellige resultater, hver gang det bliver kørt. Tre forskellige kørsler kan for eksempel give værdierne 8, 10 og 7.

Hvis man gerne vil vide, hvilke resultater der er mulige, og sandsynlighederne for de forskellige resultater, kan man køre dette program mange tusind gange. Ved at tælle antallet af gange, hvert resultat forekommer, kan man estimere sandsynlighederne. Man får dog ikke eksakte sandsynligheder: Jo flere kørsler man laver, jo bedre bliver estimerne, men man får aldrig helt præcise værdier.

Men man kan analysere programmet og finde eksakte sandsynligheder for alle udfald.

For $\max 3d10$ er de mulige udfald som følger:

Udfald	1	2	3	4	5	6	7	8	9	10
Sandsynlighed	0,1%	0,7%	1,9%	3,7%	6,1%	9,1%	12,7%	16,9%	21,7%	27,1%

Denne analyse kan laves automatisk, men kun fordi Troll er et meget simpelt sprog. Hvis man havde brugt et turingkomplet programmeringssprog til at lave terningekastene med, så ville man ikke kunne lave analysen automatisk. Samtidig gør den stærke specialisering af sproget, at de fleste almindelige måder at bruge terninger på kan laves med meget korte programmer.

Behandling af store datamængder

Søgemaskinefirmaet Google har nogle af verdens største datacentre, hvor de gemmer informationer om alverdens hjemmesider og om de søgninger, der bliver foretaget gennem Googles søgetjeneste. Alle disse data er fordelt på flere tusind computere hver med deres egne lokale diske. For at sikre sig mod maskinnedbrud findes hver bid data på flere forskellige computere på forskellige dele af det netværk, der forbinder alle computerne. Når Google skal lave statistik over disse store datamængder, for eksempel for at finde de mest almindelige søgeord, skal beregningerne effektivt fordeles over mange tusind computere, og beregningerne skal laves, så de tager højde for, at nogle af computerne bryder sammen under beregningen – noget, der sker, næsten hver gang der laves en større beregning. Det er meget komplekst at programmere denne slags fordelte beregninger i et almindeligt programmeringssprog, så Google har lavet deres eget sprog, Sawzall, til formålet.

Et program i Sawzall består af to dele: en del, der beskriver en delberegning på en enkelt bid data i isolation fra de andre dele af data, og en del, der beskriver, hvordan de delresultater, der kommer ud af de individuelle delberegninger, kombineres til et samlet resultat. Kombineringsfunktionen skal kunne tage delresultaterne i en vilkårlig rækkefølge, uden at det ændrer det endelige resultat, og uden at kombineringsfunktionen behøver at huske store datamængder undervejs. Denne begrænsning medfører, at man ikke kan lave alle tænkelige beregninger i sproget, men til gengæld er man garanteret effektiv beregning: Fordi kombineringsfunktionen kan tage delresultater i vilkårlig rækkefølge, kan den tage dem, efterhånden som de bliver færdige, og den computer, der leverede delresultatet, kan sættes i gang med en ny delberegning. Og hvis en delberegning går galt, fordi den computer, der laver delberegningen, bryder sammen, kan man genstarte delberegningen på en anden computer.

I dette tilfælde udnytter man egenskaber ved kombineringsfunktionen til effektivt at fordele beregningerne og kompensere for computersammenbrud. Det ville man ikke kunne garantere, hvis kombineringsfunktionen var skrevet i et turingkomplet sprog. Samtidig er et program i Sawzall meget kort i forhold til et tilsvarende program skrevet i et almindeligt turingkomplet sprog, da hele processen omkring fordeling af beregninger, kommunikation over netværk, håndtering af maskinnedbrud osv. er underforstået og ikke behøves at stå eksplicit i hvert Sawzallprogram. Det

herunder viste program i Sawzall finder fordelingen af Googlesøgninger på bredde og længdegrad. Den anden linje beskriver kombineringsfunktionen: Man laver en tabel over bredde- og længdegrader (lat og lon) og lægger delresultaterne sammen (sum). De tre sidste linjer beskriver behandling af de enkelte deldata: Inddata (input) er data om en enkelt søgning. Derefter udtrækkes positionsinformation ud fra IP-adressen, og man sætter et ettal ind i tabellen ved denne position. Ordet "emit" angiver, at det er det delresultat, der sendes videre til kombineringsfunktionen.

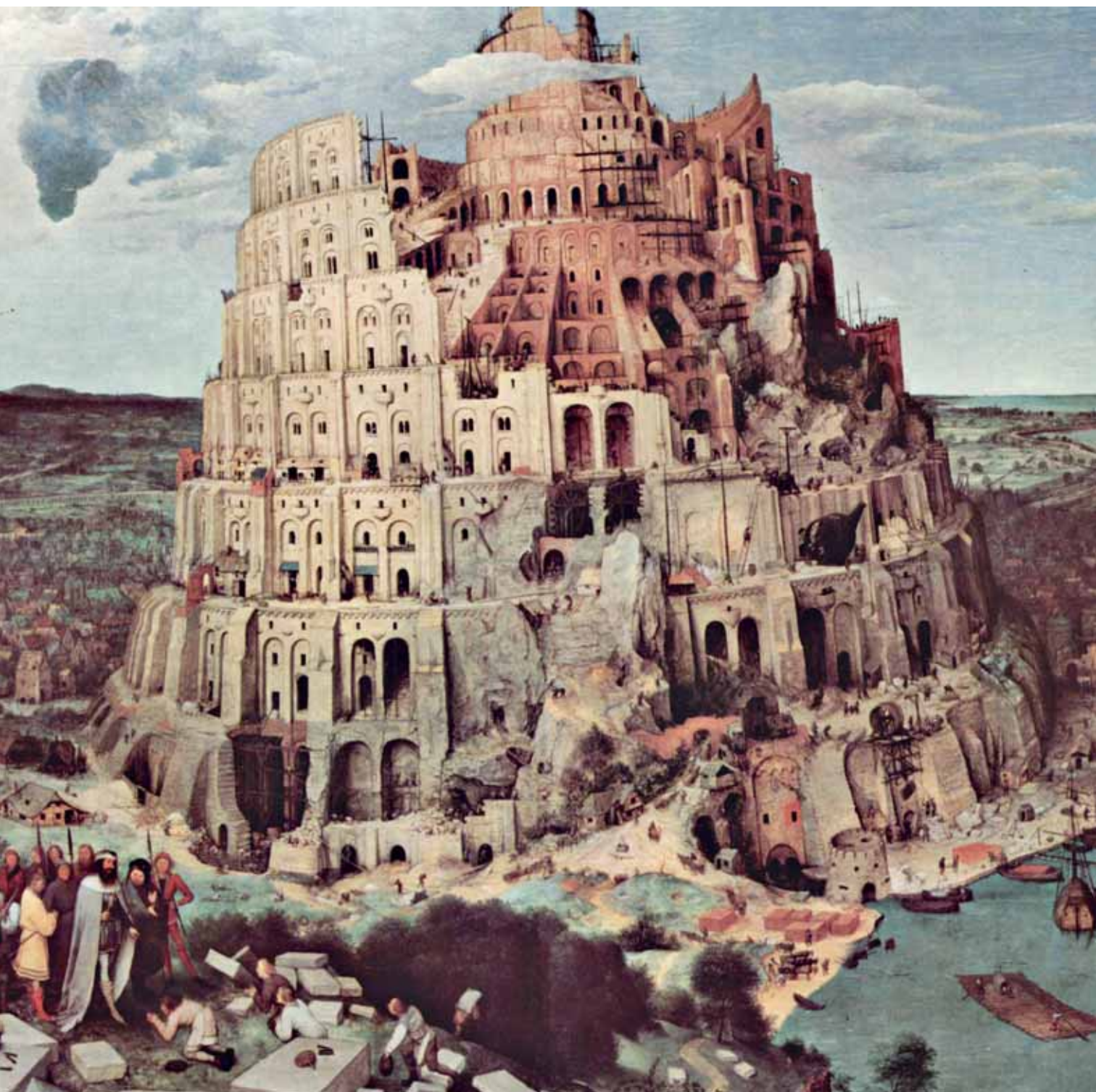
```
proto "querylog.proto"
queries_per_degree: table sum[lat: int][lon: int] of int;
log_record: QueryLogProto = input;
loc: Location = locationinfo(log_record.ip);
emit queries_per_degree[int(loc.lat)][int(loc.lon)] <-1;
```

Tabellen `queries_per_degree` kan vises som et billede: Antallet af forespørgsler fra et bestemt sted vises som farven i et billedpunkt: Jo mørkere punktet er, jo flere forespørgsler. Et billede svarende til forespørgslerne i et døgn i august 2003 er vist herunder.



Reversible sprog

En egenskab, som et svagt sprog kan have, men som et turingkomplet sprog ikke har, er reversibilitet: Evnen til at køre et program baglæns, så man fra uddata kan finde det inddata, der giver dette uddata. Reversibilitet kræver, at der for hvert muligt uddata kun er et muligt inddata, der giver dette uddata. Så et program, der som inddata tager to tal og som uddata giver deres sum, er ikke reversibelt – der er mange forskellige inddata, der giver samme uddata. Men et program, der som inddata tager to tal og som uddata giver *både* deres sum og deres differens, kan være reversibelt: Hvis man kender uddata, er der kun ét muligt inddata, for hvis $a = x+y$ og $b = x-y$, så er $x = (a+b)/2$ og $y = (a-b)/2$.



The Tower of Babel, Pieter Bruegel, 1563 - Kunsthistorisches Museum, Vienna. Historien om Babelstårnet illustrerer mangfoldigheden af sprog.



*Tendensen
går i retning
af specialise-
rede sprog*

I et reversibelt programmeringssprog kan man kun skrive reversible programmer, og man kan implementere sproget på en reversibel computer: en computer, der kun kan lave reversible beregninger

En konsekvens af termodynamikkens anden lov er, at en reversibel computer kan laves, så den bruger mindre energi end en ikke-reversibel computer. Så til formål, hvor lavt energiforbrug er vigtigt, kan reversible computere og sprog være interessante. Indtil videre er det dog fremtidsmusik, da energieffektive reversible computere endnu kun er på tegnebrættet.

Men reversible sprog har en anden fordel. Ofte har man brug for at lave beregninger både forlæns og baglæns: Hvis man for eksempel komprimerer data til en kompakt form, vil man gerne kunne genskabe det oprindelige data, så man laver i reglen to programmer: et til at komprimere data og et andet til at genskabe det originale data fra det komprimerede. Hvis man skriver komprimeringsprogrammet i et reversibelt sprog, behøver man ikke at skrive det andet program: Man kan bare køre komprimeringsprogrammet baglæns i stedet. I de reversible programmeringssprog, der findes i dag, kan det dog være vanskeligere at skrive dette ene program, end det vil være at skrive de to indbyrdes omvendte programmer.

Det kan bemærkes, at reversible beregninger er en forudsætning for kvantecomputere, der udnytter kvantemekanikkens superpositionsbegreb til at lave beregninger på mange værdier samtidig. De beregninger, der kan laves på en kvantecomputer, er dog kun en delmængde af alle reversible beregninger.

Opsummering

Der findes tusindvis af forskellige programmeringssprog. De fleste kan beregne det samme: alle beregninger, der kan automatiseres. Det kan blandt andet være svært at sige, at et sprog er bedre end et andet, uden at specificere en meget præcis kontekst for sammenligningen.

Der kan være grunde til at bruge programmeringssprog, der ikke kan beregne alting: Begrænsningerne kan give andre fordele såsom ressourcegarantier, analyserbarhed og parallel beregning.

Der kommer hele tiden nye programmeringssprog til, og tendensen går i retning af specialiserede sprog. Og selv folk, der ikke betragter sig som it-arbejdere, vil i stigende grad bruge simple programmeringssprog såsom regneark eller stærkt specialiserede sprog. ♦

Læs mere

Alan Turing og turingmaskiner: http://en.wikipedia.org/wiki/Alan_turing

Alonzo Church og lambdakalkulen: http://en.wikipedia.org/wiki/Alonzo_Church

Troll: <http://topps.diku.dk/torbenm/troll.msp>

Sawzall: <http://labs.google.com/papers/sawzall.html>

Reversible beregninger: http://en.wikipedia.org/wiki/Reversible_computing

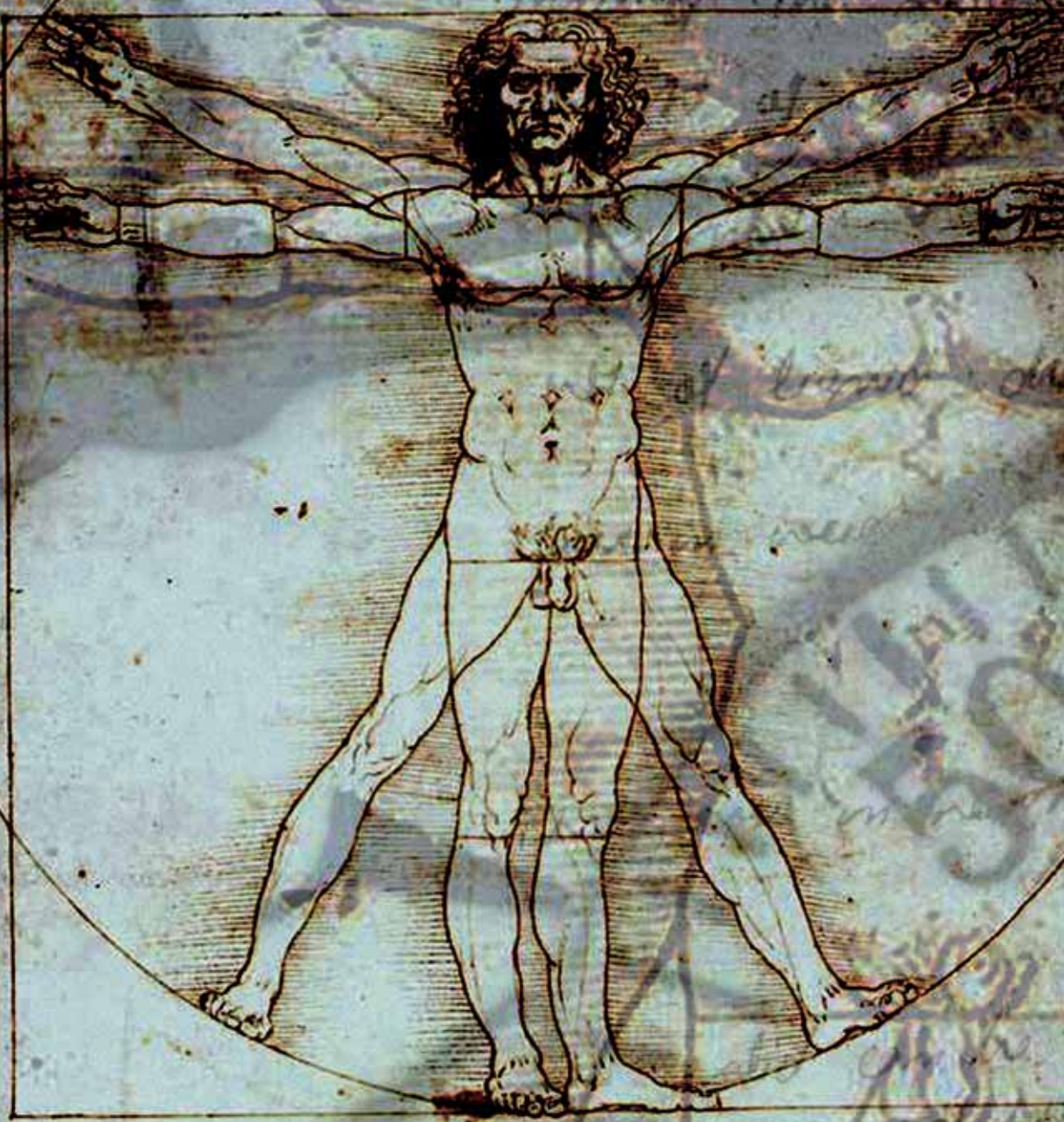
TORBEN ÆGIDIUS MOGENSEN



Torben Ægidius Mogensen har siden 1994 været lektor og siden 2007 studieleder på Datalogisk Institut på Københavns Universitet. Torben er ph.d. fra 1989 og har modtaget en undervisningspris fra Det Naturvidenskabelige Fakultet i 1998. Han har desuden været med til at akkreditere uddannelser i Estland.

Torben har bl.a. skrevet lærebogen "Basics of Compiler Design" (2000, 2010), som han anvender i sin under-

visning inden for samme område på DIKU. Som en del af APL-gruppen (algoritmer og programmeringssprog) arbejder Torben med automatisk programanalyse og -transformation, oversættere og domænespecifikke sprog. Af udvalgte artikler kan nævnes: "Troll: A Language for Specifying Dice-Rolls", ACM Symposium on Applied Computing 2009, "Semi-Inversion of Functional Parameters", Partial Evaluation and Program Manipulation 2008, og "Linear-Time Self-Interpretation of the Pure Lambda Calculus", Journal of Higher-Order and Symbolic Computation 2000.



Handwritten text in Italian, likely a transcription or commentary related to the drawing above. The text is written in a cursive script and is arranged in several lines. The first line is a heading or title, followed by several paragraphs of text. The text is written on a piece of paper that is slightly aged and has some staining. The handwriting is clear and legible, though some words are difficult to decipher due to the cursive style. The text appears to be a detailed description or analysis of the figure and the geometric shapes used in the drawing, possibly discussing the proportions of the human body and the relationship between the figure and the circle and square.



DEN DIGITALE TEKNIKS MULIGHEDER

solid = Original weight (weight in air)
- weight of liquid displaced

$$\begin{aligned}\text{The solid} &= \rho_s V_s g - \rho_f V_s g \\ &= (\rho_s - \rho_f) V_s g\end{aligned}$$

the air & then in the
the loss in weight of the
can calculate ρ_s if



Den digitale tekniks muligheder

.....
Af redaktionen
.....

Den digitale verden udvikler sig nærmest eksplosivt, og der er tilsyneladende ingen grænser for de muligheder, som den giver os mennesker. Mobiltelefonens udvikling frem til at blive et helt lille multimediecenter i lommeformat er et lysende eksempel på denne udvikling. Men selvfølgelig er der grænser for alt. I det følgende vil vi kort se på it-system-tankegangen og de muligheder, den frembyder.

Den digitale verden kan anskues som værende opbygget af et utal af it-systemer, hvor de enkelte programmer fungerer som byggesten. Det kan være mindre programmer, som løser en velafgrænset opgave som eksempelvis styring af cursoren på laptoppen, eller større programkomplekser som eksempelvis Microsoft Office-pakken.

Når man skal konstruere et it-system, skal man have en vision – en idé om, hvad det pågældende it-system skal kunne. Man har altså en forestilling om det fænomen, man ønsker at beskrive, som kan være et fænomen fra den fysiske verden eller en rent mental forestilling. Herefter skal man udarbejde en systematisk og ordnet beskrivelse af det betragtede fænomen, og man får hermed opstillet en model. Modellen er ikke en direkte afbildning af det betragtede fænomen, men en forenklet beskrivelse, hvor man har valgt at fremhæve centrale elementer af fænomenet og udeladt mindre betydende elementer.

I arbejdet med at formulere modellen vil man ofte trække på en faglig viden og faglige teorier. Vil man eksempelvis konstruere et system, som kan afbilde aktiviteterne i den menneskelige hjerne, vil man trække på hjerneforskernes viden, og vil man lave et system, som kan komme med forudsigelser om vejret, må man nødvendigvis trække på meteorologers viden. Ved at opstille sådanne modeller får man en overordnet forståelse af det fænomen, man betragter, og man får en idé om, hvilke dele der skal implementeres i it-systemet. Den opstillede model er ikke selv en del af den digitale verden, men en opskrift på, hvorledes man skal opbygge it-systemet.

Når man nu skal omsætte den opstillede model til et kørende it-system og dermed en del af den digitale verden, opstår der nye begrænsninger. Det kræver nemlig, at modellen skal kunne beskrives i præcise og eksakte data, dvs. ned til mindste detalje. For at opnå den nødvendige præcision er det hensigtsmæssigt at formulere sig i det matematiske sprog, således at vi får en matematisk beskrivelse af den opstillede model. Det vil for det meste være nødvendigt at bruge tilnærmede beskrivelser. Man kan sige, at it-systemet repræsenterer endnu en forenkling af det betragtede fænomen. Igennem hele denne proces må man til stadighed holde sig for øje, om det system, man er ved at konstruere, opfylder den oprindeligt formulerede vision.

Den digitale verden består altså af forenklede beskrivelser af fænomener fra den virkelige verden, og der er ubegrænsede muligheder for at opbygge sådanne beskrivelser og implementere dem som en del af den digitale verden. Hvorvidt det enkelte menneske så kan bruge disse systemer, er imidlertid en ganske anden sag. For at vurdere anvendeligheden af de enkelte it-systemer skal man have hele den ovenfor skitserede proces for øje, idet man skal vurdere gyldigheden og anvendeligheden af det enkelte systems resultater. Har man eksempelvis udviklet et it-system, som kan udregne kvinders risici for at få brystkræft, skal man være opmærksom på, at der er tale om gennemsnitsbetragtninger, som gælder for befolkningen som helhed, men som ikke nødvendigvis siger noget om den enkelte kvindes risiko for at få brystkræft.

Menneskers drømme og ønsker har været og er stadig i høj grad styrende for de it-systemer, man ønsker at realisere. Der er dog ikke garanti for, at den nødvendige viden er til stede. Således har man siden midten af 70'erne gentagne gange proklameret konstruktionen af den intelligente computer, men dette mangler vi stadig at se. Vi har simpelthen ikke den nødvendige viden.

I det følgende går vi først tæt på fænomenet modellering, idet Kenny Erleben giver et indblik i, hvordan man simulerer eller modellerer den menneskelige bevægelse – til brug for robotter eller animationer i film.

Herefter går vi over til en helt anden mulighed, nemlig at lade computeren gøre arbejdet lettere for os ved at optimere. David Pisinger fortæller om, hvordan forskning i algoritmer og optimering kan skabe bæredygtig it.

Mads Nielsen kommer ind på nogle spændende tekniske anvendelsesområder inden for bl.a. medicinsk forskning, nemlig med udgangspunkt i fænomenet machine learning, der handler om at benytte computeren som et redskab til automatisk læring.

Endelig giver Morten Kringelbach os en indføring i filosofien bag udviklingen af de såkaldt bevidste eller intelligente computere – fra en hjerneforskners perspektiv. ❖

Simulering og modellering af robotter og mennesker

.....
Af **Kenny Erleben**, DIKU
.....

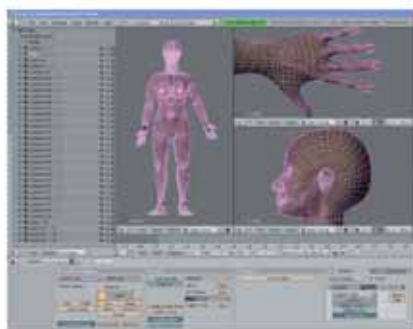
Simulering og modellering af robotter og mennesker er vel nok et af de ældste emner inden for datalogien. Simulering anvendes i dag i mange sammenhænge, fx når animatoren skal få en figur til at bevæge sig i et computerspil eller en film, eller når ingeniøren skal lave en opskrift på, hvordan en robot skal bevæge sig. Datalogen kommer således både animatoren og ingeniøren til hjælp, idet begge bruger computerprogrammer til at modellere og simulere hhv. robotens og menneskets bevægelse. Det er nemlig datalogen, som har skabt de modeller og algoritmer, der bruges i computerprogrammerne.

Ovenstående er et godt eksempel på, at datalogi er et tværfagligt fag. Datalogen bygger modeller af de problemer, der skal løses. Som led i problemløsningen kommer datalogen typisk i nærkontakt med matematik og andre fag såsom biologi, kemi og fysik. I denne artikel gives et eksempel på, hvordan en datalog modellerer ved hjælp af matematikken. Vi vil kigge på fænomenet **invers kinematik**. Invers kinematik danner grundlag for mange værktøjer, der bruges inden for faglige felter som robotik, biomekanik, computergrafik og computeranimation. Eksempler kan ses i figur 1.

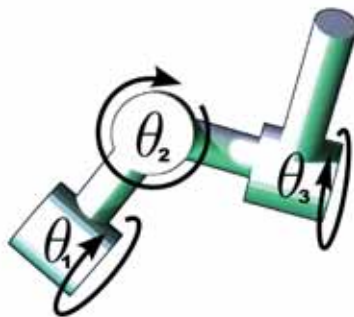
Invers kinematik bruges til at beregne, hvordan et menneske eller en robot skal placere sig for at kunne nå en ting som fx et stykke værktøj eller en kop kaffe. Før man kan begynde at løse det inverse kinematiske problem, må man bygge en model, der udregner, hvor hænderne og fødderne vil befinde sig, afhængigt af hvor meget man bøjer knæ og albuer. Når vi har denne model, kan vi løse problemet, nemlig: Givet hvor hænder og fødder er, hvor meget skal vi så bøje albuer og knæ for at kunne nå genstanden? Denne tilgang illustrerer den inverse problemstilling, også betegnet invers kinematik som er en matematisk beskrivelse af bevægelse, uafhængigt af bevægelsens årsag. Man opstiller bevægelsesligninger, der beskriver forholdene. Det centrale begreb er position opfattet som en funktion af tiden.

En matematisk model af et menneske

Et menneske er en meget kompleks organisme. Hvis vi ønsker kun at beskrive, hvordan et menneske bevæger sig i en afgrænset sammenhæng, kan vi bruge en simplere model. Hvis vi kun vil vide, hvorvidt en person har hånden oppe i luften, eller hvor meget en arm eller et knæ er bøjet, kan vi nøjes med at modellere et menneske som et skelet af stive stænger. Stængerne er sat sammen dér, hvor et menneske har et led. Overarmen og underarmen er forbundet af albueledet, som rent mekanisk kan opfattes som et hængselled. Vi kan nu beskrive, hvordan underarmen er placeret i forhold til overarmen ved at beskrive vinklen imellem overarmen og underarmen. Tilsvarende kan vi opfatte et skulderled eller et hoftelid som et kugleled, der hæfter overarmen fast på overkroppen og overlåret fast på underkroppen. Figur 2 viser et eksempel.



(a)



(b)



(c)



(d)

Figur 1: Forskellige eksempler på brug af modellering og simulering af mennesker. I (a) ses et animationsprogram, der blev brugt til at lave filmen Big Buck Bunny med. I (b) ses, hvordan en robot kan beskrives ud fra sin kinematik, hvorimod (c) viser en invers kinematik-software udviklet på DIKU. Endelig ses i (d) et eksempel, hvor softwaren fra (c) bruges til at opfange den menneskelige bevægelse med et kamera.

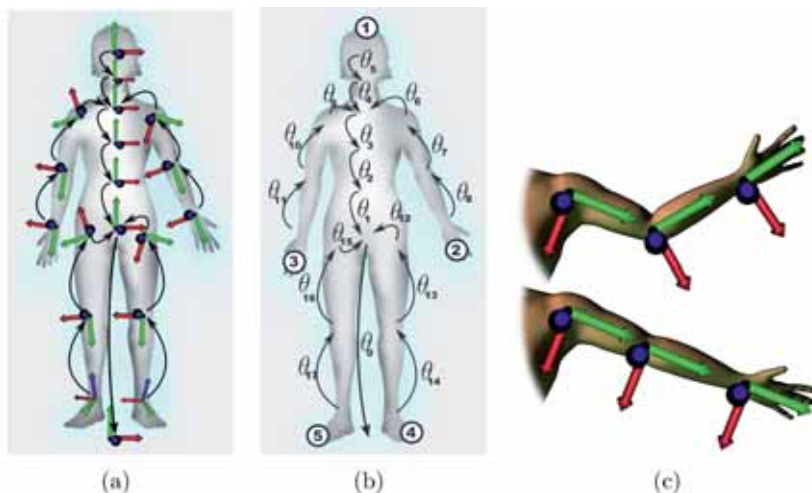


(a)



(b)

Figur 2: I (a) ses et eksempel på, hvordan et menneske kan gengives som en simpel model af stive stænger sammensat af forskellige slags bevægelige led. Bemærk i (b), at modellen også passer godt på en robot.



Figur 3: I (a) ses et eksempel på, hvordan koordinatsystemer og transformationer svarer til figur 2(a). I (b) ses en indeksering af led og kæder af led. I (c) ses et detaljeret eksempel på en arm, og hvordan koordinatsystemer kan placeres i den. Bemærk, hvordan koordinatsystemerne ændres, når armen strækkes.

Som det ses, er stangmodellen som skabt til også at beskrive fx robotter. Det problem, vi ønsker at løse, er: Givet albuevinklen og skuldervinklerne, hvor er hånden så placeret? Med andre ord, hvad er koordinaterne af modellens hænder og fødder, afhængigt af hvor meget den bøjer eller strækker sine led?

For at kunne regne effektivt med vores stangmodel af et menneske må vi udtrykke den ved hjælp af matematik. Vi benytter os af koordinatsystemer og transformationer imellem koordinatsystemer, idet vi placerer et koordinatsystem i hvert bevægelsesled af vores stangmodel, hvor de stive stænger beskriver, hvordan koordinatsystemerne er placeret i forhold til hinanden. Det er stangen, der beskriver, hvordan vi skal transformere et koordinatsystem i den ene ende af stangen, således at det transformerede koordinatsystem er sammenfaldende med koordinatsystemet i den anden ende af stangen. Figur 3 illustrerer idéen med at bruge koordinatsystemer og transformationer.

Vi introducerer nu en bestemt rækkefølge af vores koordinatsystemer og en entydig retning af vores koordinatstransformationer. Først udvælger vi en stang i vores model, som vi kalder roden. I princippet kan man vælge vilkårligt, men for mennesker vil man typisk vælge den stang, som svarer til bækkenet på et menneske. Vi definerer nu en rækkefølge ved rekursivt at følge de stænger, der er forbundet til roden. Dette gør vi ved hele tiden at følge en stang, som vi ikke har set før, og som er fastgjort til en stang, vi allerede har kigget på. Når vi ikke kan komme videre fra en stang, har vi fundet det, der kaldes en slutstang. På denne måde skaber vi en mængde sammenkædede stænger, der alle går fra en rod til en slutstang. Med koordinatstransformationerne beskriver vi, hvordan et koordinatsystem i retning mod slutstangen kan transformeres til koordinatsystemet i retning af roden.

Når mennesket bevæger sig rundt i verden, sker det ved, at vi flytter roden rundt i verden. Vi tilføjer derfor et ekstra koordinatsystem til at hjælpe os med at huske, hvor modellen er henne i forhold til verden. Det ekstra koordinatsystem angiver, hvor nulpunktet i verden er.

Vores valgte definition af rækkefølgen betyder at vores model danner en træstruktur på samme måde som et familietræ. Roden svarer til ens tip-tip...-oldeforældre, som har startet det hele, og slutstængerne svarer til den seneste generation af børn i familien. De mellemliggende stænger kan sammenlignes med deres forældre og så videre. Med alle disse valg opnås den fordel, at vi ikke behøver at huske koordinatsystemerne, når vi senere skal bruge modellen i praksis; hver stang behøver nemlig kun at huske koordinattransformationen til sine egne forældre. Figur 4 illustrerer træstrukturen.



Transformationerne består af rotationer og translationer



Figur 4: Af illustrationen ses det, hvordan stangmodellen svarer til en træstruktur. Bemærk, at hver knude i træet blot skal huske, hvem dens forældre er for at konstruere det fulde træ.

Hver kæde af stænger svarer til en vej fra roden af træet til et af bladene. Vi er nu nået til den sidste beslutning i vores model, nemlig hvordan vi skal repræsentere koordinattransformationerne. Vi bruger matematiske udtryk for at opnå tilstrækkelig præcision. Transformationerne består af rotationer og translationer, og vi benytter en vektor- og matrixnotation til at udtrykke disse; se også eksemplet på næste side.

Eksempel på vektor- og matrixnotation samt forskellige regneoperationer

En matrix er en tabel af talværdier. Størrelsen af tabellen angiver dimensionen af matricen. Når vi regner med formler, bruger vi symboler i stedet for talværdier. Dvs., vi skriver:

$$\mathbf{A} = \begin{bmatrix} A_{11} & \dots & A_{1N} \\ \vdots & \ddots & \vdots \\ A_{M1} & \dots & A_{MN} \end{bmatrix} \quad (1)$$

Her vil \mathbf{A} altså være en matrix med dimensionerne $M \times N$. Talværdien i den i 'te række og den j 'te kolonne skrives som A_{ij} . Har en matrix kun én kolonne, kaldes den for en vektor. Ligesom vi kan regne med tal, kan vi også regne med matricer og vektorer. To matricer af samme dimension kan lægges sammen og trækkes fra hinanden elementvis, dvs:

$$\mathbf{C} = \mathbf{A} - \mathbf{B} \quad \text{hvor} \quad C_{ij} = A_{ij} - B_{ij} \quad (2a)$$

$$\mathbf{C} = \mathbf{A} + \mathbf{B} \quad \text{hvor} \quad C_{ij} = A_{ij} + B_{ij} \quad (2b)$$

Matricer kan også transponeres, hvilket betyder, at vi spejler dem på skrå ned langs deres midte. Det skriver vi som:

$$\mathbf{C} = \mathbf{A}^T \quad \text{hvor} \quad C_{ij} = A_{ji} \quad (3)$$

Endelig kan to matricer \mathbf{A} og \mathbf{B} med dimensionerne $M \times K$ og $K \times N$ også multipliceres med hinanden:

$$\mathbf{C} = \mathbf{AB} \quad \text{hvor} \quad C_{ij} = \sum_{k=1}^K A_{ik} B_{kj} \quad (4)$$

Bemærk, at \mathbf{C} vil have dimensionen $M \times N$. Endelig kan en matrix multipliceres med et enkelt tal a :

$$\mathbf{C} = a\mathbf{A} \quad \text{hvor} \quad C_{ij} = aA_{ij}. \quad (5)$$

Som en lille testopgave kan læseren prøve at udregne følgende lille problem. Givet

$$\mathbf{H} = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \quad \text{og} \quad \mathbf{p} = \begin{bmatrix} 1 \\ 2 \end{bmatrix} \quad (6)$$

hvad er så værdien af vektoren \mathbf{q} ?

$$\mathbf{q} = \frac{1}{2} \mathbf{H}^T \mathbf{p} \quad (7)$$

I et to dimensionelt koordinatsystem er et punkt $\mathbf{p} = [x \ y]$ givet ved to koordinater x - og y koordinaterne. En rotation med vinklen θ af punktet \mathbf{p} kan skrives som

$$\mathbf{p}_{\text{rot}} = \begin{bmatrix} x_{\text{rot}} \\ y_{\text{rot}} \end{bmatrix} = \underbrace{\begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix}}_{\mathbf{R}(\theta)} \begin{bmatrix} x \\ y \end{bmatrix} = \mathbf{R}(\theta)\mathbf{p} \quad (8)$$

En translation kan skrives matematisk som

$$\mathbf{p}_{\text{trans}} = \begin{bmatrix} x_{\text{trans}} \\ y_{\text{trans}} \end{bmatrix} = \begin{bmatrix} x \\ y \end{bmatrix} + \underbrace{\begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix}}_{\mathbf{t}} = \mathbf{p} + \mathbf{t} \quad (9)$$

Vi kan kombinere både translation og rotation i en fælles matematisk notation ved hjælp af et lille matematisk trick

$$\begin{bmatrix} \mathbf{p}_{\text{rot}} + \mathbf{p}_{\text{trans}} \\ 1 \end{bmatrix} = \underbrace{\begin{bmatrix} \mathbf{R}(\theta) & \mathbf{t} \\ 0 & 1 \end{bmatrix}}_{\mathbf{T}(\theta)} \begin{bmatrix} \mathbf{p} \\ 1 \end{bmatrix} \quad (10)$$

Tricket består i at udvide alle vores punkter med en ekstra koordinat, som altid har værdien 1. Herved kan vi skrive en koordinattransformation bestående af både en translation og rotation som en matrixmultiplikation med matricen $\mathbf{T}(\theta)$. For hver af koordinattransformationerne svarende til pilene i figur 3 har vi præcis en matrix $\mathbf{T}(\theta)$. Til at hjælpe os med at bevare overblikket vil vi benytte indeksnotation, således at matricen $\mathbf{T}_{ij}(\theta_j)$ svarer til koordinattransformationen mellem den j 'te stang fra den i 'te række/kæde og dens forældre.

Bemærk, at θ_j er vinklen mellem den j 'te stang og dens forældre. Vi kan nu løse vores oprindelige problem: Givet alle vinkelværdier θ_j 'er, hvor befinder hænder og fødder sig så henne i verden? Rent matematisk kan dette svar skrives som:

$$\underbrace{\begin{bmatrix} \mathbf{p}_{\text{hoved}} \\ \mathbf{p}_{\text{venstre hånd}} \\ \mathbf{p}_{\text{phøjre hånd}} \\ \mathbf{p}_{\text{venstre fod}} \\ \mathbf{p}_{\text{phøjre fod}} \end{bmatrix}}_{\mathbf{g}} = \underbrace{\begin{bmatrix} (\mathbf{T}_{1,0}(\theta_0)\mathbf{T}_{1,1}(\theta_1)\mathbf{T}_{1,2}(\theta_2)\mathbf{T}_{1,3}(\theta_3)\mathbf{T}_{1,4}(\theta_4)\mathbf{T}_{1,5}(\theta_5)) \\ (\mathbf{T}_{1,0}(\theta_0)\mathbf{T}_{1,1}(\theta_1)\mathbf{T}_{1,2}(\theta_2)\mathbf{T}_{1,3}(\theta_3)\mathbf{T}_{2,6}(\theta_6)\mathbf{T}_{2,7}(\theta_7)\mathbf{T}_{2,8}(\theta_8)) \\ (\mathbf{T}_{1,0}(\theta_0)\mathbf{T}_{1,1}(\theta_1)\mathbf{T}_{1,2}(\theta_2)\mathbf{T}_{1,3}(\theta_3)\mathbf{T}_{3,9}(\theta_9)\mathbf{T}_{3,10}(\theta_{10})\mathbf{T}_{3,11}(\theta_{11})) \\ (\mathbf{T}_{1,0}(\theta_0)\mathbf{T}_{4,12}(\theta_{12})\mathbf{T}_{4,13}(\theta_{13})\mathbf{T}_{4,14}(\theta_{14})) \\ (\mathbf{T}_{1,0}(\theta_0)\mathbf{T}_{5,12}(\theta_{12})\mathbf{T}_{5,13}(\theta_{15})\mathbf{T}_{5,16}(\theta_{17})) \end{bmatrix}}_{\mathbf{F}(\theta_1, \dots, \theta_n)} \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 1 \end{bmatrix}. \quad (11)$$



En animeret 3d-model af det menneskelige skulderkompleks. Datasættet stammer fra et forskningsprojekt om muskelsimulation.

Her har vi udnyttet, at hænder, fødder og hoved befinder sig i nulpunktet af deres eget koordinat-system, hvilket skrives som $[0 \ 0 \ 1]^T$. Vi har også udnyttet, at kæderne deler nogle af deres transformationer, f.eks er $\mathbf{T}_{1,1}(\theta_1) = \mathbf{T}_{2,1}(\theta_1)$ og så fremdeles. Vores idealiserede matematiske model af et menneske kan nu skrives kompakt i en ligning som:

$$\mathbf{g} = \mathbf{F}(\theta_1, \dots, \theta_n) \quad (12)$$

Det inverse problem

Vores model fra ligning (12) kan bruges til at finde ud af, hvor hænder og fødder er henne. Det inverse problem består i at finde ud af, hvordan knæ og albuer skal bøjes for at få figurens hænder til at nå et mål, fx at samle en kop op fra et bord. Det er af stor værdi at kunne løse dette problem, idet det indebærer, at animatoren så blot behøver at fortælle computerprogrammet, at hans figur skal samle en kop op - så finder programmet selv ud af, hvordan figuren skal stå. På samme måde kan en robot bruge det inverse problem til at finde ud af, hvordan den skal styre sine motorer, så den kan samle et værktøj op.

Rent matematisk kan vi beskrive vores problem som: Givet \mathbf{g} ønsker vi at finde alle θ_j 'er, således at:

$$\underbrace{[\theta_1 \ \dots \ \theta_n]^T}_{\boldsymbol{\theta}} = \mathbf{F}^{-1}(\mathbf{g}) \quad (13)$$

hvor $\mathbf{F}^{-1}(\bullet)$ er den inverse funktion af $\mathbf{F}(\bullet)$. Det vil sige, at $\mathbf{g} = \mathbf{F}(\mathbf{F}^{-1}(\mathbf{g}))$.

Vores problem er altså nu at finde den inverse funktion $\mathbf{F}^{-1}(\bullet)$. De fundne θ_j -værdier kaldes for en løsning og skrives som $\boldsymbol{\theta}^*$. Stjernenotationen angiver, at der er tale om en løsning.

Umiddelbart er dette et temmelig svært problem at løse, og det er ikke altid praktisk muligt at finde den inverse funktion, da den ikke nødvendigvis eksisterer eller er entydig. Hvad værre er, det er ikke sikkert, at det ønskede \mathbf{g} rent faktisk kan nås. I sådanne tilfælde forsøger vi i stedet at finde de θ_j 'er, som bringer os tættest muligt på \mathbf{g} . Så i stedet for at løse ligning (13) løser vi et lidt andet problem, der om ikke andet har de samme løsninger som problemet i ligning (13). Som alternativ ønsker vi at finde de θ_j 'er, som giver den mindst mulige afstand mellem det givne \mathbf{g} og $\mathbf{F}(\boldsymbol{\theta}_1, \dots, \boldsymbol{\theta}_n)$. Dette kan skrives som:

$$\boldsymbol{\theta}^* = \underset{\boldsymbol{\theta}}{\operatorname{argmin}} \|\mathbf{g} - \mathbf{F}(\boldsymbol{\theta})\| \quad (14)$$

Vi ønsker at minimere objektfunktionen med hensyn til argumentet $\boldsymbol{\theta}$. Afstandsfunktionen af en n -dimensionel vektor \mathbf{v} er defineret som:

$$\|\mathbf{v}\| = \sqrt{v_1^2 + \dots + v_n^2} \quad (15)$$



Det er ikke altid praktisk muligt at finde den inverse funktion

Denne funktion er altid ikke-negativ og simpelt monoton, og da værdien nul af vores objektfunktion giver os den løsning, vi leder efter, kan vi kvadrere vores objektfunktion for at slippe af med kvadratroden. Herved kan vi skrive:

$$\|\mathbf{v}\|^2 = v_1^2 + \dots + v_n^2 = \mathbf{v}^T \mathbf{v} \quad (16)$$

Vi kan ligeledes multiplicere vores objektfunktion med en vilkårlig positiv talværdi uden at ændre på løsningen. Vores endelige problemformulering bliver derfor:

$$\theta^* = \underset{\theta}{\operatorname{argmin}} \underbrace{\frac{1}{2} (\mathbf{g} - \mathbf{F}(\theta))^T (\mathbf{g} - \mathbf{F}(\theta))}_{f(\theta)} \quad (17)$$



Vi er nødt til at omforme problemet til noget, computeren forstår

Her er $f(\theta)$ vores endelig objektfunktion, som vi vil bruge. Den har de samme løsninger som objektfunktionen i ligning (14). Fordelen ved f er, at den er nemmere at regne med og giver nogle mere simple ligninger.

Den diskrete løsning

Modellerne fra ligningerne (12), (13) og (17) er, hvad man kalder for idealiseringer. De er pæne matematiske modeller, der forsøger at beskrive, hvordan mennesker i den rigtige verden bevæger sig. En del af idealiseringen består i at skære ting væk, som ikke er væsentlige for det problem, vi ønsker at løse. Fx er hårfarven irrelevant i forhold til at finde ud af, hvor ens hænder befinder sig. Vi kan ikke umiddelbart indtaste en matematisk ligning som (17) i en computer, men er nødt til at omforme problemet til noget, som computeren forstår. Vi er også nødt til at beskrive, hvordan computeren rent faktisk skal løse problemet. Denne næste fase af modelleringsprocessen kaldes for en diskretisering.

Vi anvender en iterativ metode til at løse optimeringsproblemet i ligning (17). Først får vi givet et startgæt for θ^* skrevet som θ^0 . Vi vil så forsøge at beregne et nyt og bedre gæt θ^1 , således at $f(\theta^1) < f(\theta^0)$. Hvis θ^1 ikke minimerer $f(\theta^1)$, forsøger vi at beregne endnu et gæt θ^2 og så fremdeles. Vi bliver ved med at gætte, indtil vi finder et gæt, vi kan lide. Vi kan tænke på $f(\theta)$ som en højdefunktion af et landskab på nøjagtig samme måde som et højdekort i et atlas over jordkloden. Når vi står med et eller andet gæt θ^k , ønsker vi at finde en måde at ændre θ^k på til θ^{k+1} , således at:

$$f(\theta^{k+1}) < f(\theta^k) \quad (18)$$

Vi ved, at gradienten af en funktion i punktet θ^k , $\nabla f(\theta^k)$ altid peger i den retning, som funktionen vokser mest i. Se faktaboks for information om gradienten af en funktion.

Definition af en gradient

Har vi en funktion

$$f(\theta) = a\theta^2 + b\theta + c \quad (19)$$

hvor a , b og c er vilkårlige men konstante talværdier, kan vi differentiere f med hensyn til θ .
Det skriver vi som:

$$\frac{\partial f(\theta)}{\partial \theta} = 2a\theta + b \quad (20)$$

Har vi imidlertid en funktion, som afhænger af flere argumenter, kaldes det en vektorfunktion og vi skriver

$$f(\theta_1, \theta_2, \dots, \theta_n) = f(\theta) \quad \text{hvor} \quad \theta = [\theta_1 \quad \theta_2 \quad \dots \quad \theta_n]^T \quad (21)$$

Vi kan nu differentiere f med hensyn til et af dens argumenter θ_i , dvs vi udregner:

$$\frac{\partial f(\theta)}{\partial \theta_i} \quad (22)$$

Gør vi dette for alle argumenter og samler resultaterne i en vektor, så får vi det, man kalder for gradienten af funktionen. Vi skriver dette som:

$$\nabla f(\theta) = \left[\frac{\partial f(\theta)}{\partial \theta_1} \quad \dots \quad \frac{\partial f(\theta)}{\partial \theta_n} \right]^T \quad (23)$$

Rent intuitivt kan gradienten opfattes som en vektor, der peger i den retning, som funktionen vil vokse hurtigst i.

Når vi skal finde en bedre værdi end θ^k , kan vi kigge i den modsatte retning af gradienten. Man kan tænke på gradienten som en måling af, hvor meget landskabet skråner, lige der hvor man står, dvs. det er en lokal måling, der kun gælder i et lille område omkring ens målepunkt. Man skal derfor passe på ikke at gå for langt i den modsatte retning af gradienten. Derfor multiplicerer vi vores gradients skridt med en tilpas lille positiv værdi τ . Vores nye bud kan findes som:

Vores eneste problem er nu at finde værdien τ i ligning (24). Det gør vi ved at gætte os frem til en værdi på en intelligent måde. Vi starter med at sætte $\tau = 1$ og tester så, om dette vil give os en θ^{k+1} -værdi, der opfylder ligning (18). Vi ved, at gradienten er en lokal måling, så vi skal bare gøre

$$\theta^{k+1} = \theta^k - \tau \nabla f(\theta^k) \quad (24)$$

skridtlængden kortere og kortere, indtil vi er i et område, der er lokalt nok til, at gradienten giver en god måling. Vi halverer værdien af τ og tester igen ligning (18). Sådan fortsætter vi iterativt, indtil vi har fundet en passende τ -værdi.

Vi kan nu opskrive den fulde algoritme, som computeren bruger til at finde en løsning. Til at starte med gives en værdi for målene og et startgæt på vinklerne θ^o ,

```
1 :  $k \leftarrow 0$ 
2 : while  $\|\nabla f(\theta^k)\| > \varepsilon$  do
3 :    $\tau \leftarrow 1$ 
4 :   while  $f(\theta^k - \tau \nabla f(\theta^k)) \geq f(\theta^k)$  do
5 :      $\tau \leftarrow \frac{1}{2}\tau$ 
6 :   end
7 :    $\theta^{k+1} \leftarrow \theta^k - \tau \nabla f(\theta^k)$ 
8 :    $k \leftarrow k + 1$ 
9 : end
```

Læg især mærke til testen, der udføres i linje 2 af algoritmen. I denne linje testes der for, om gradienten i det punkt, man står i, er tilpas tæt på nul. Her svarer ε til en lille positiv talværdi, som er givet. Det er nødvendigt at gennemføre en sådan test i en computer, fordi afrundings- og præcisionsfejl gør, at man ikke vil kunne få præcis værdien nul i en udregning. Hvis gradienten af en funktion er nul, betyder det, at vi har fundet et ekstremum, dvs. enten et minimumspunkt, saddepunkt eller maksimumspunkt. Da det netop er et minimumspunkt, der skal være vores løsning, vil algoritmen altså slutte med at gætte på en θ -værdi, som kan være et minimum. Vores algoritme er nu færdigdesignet og klar til at blive programmeret, så den kan bruges i et computerprogram, eksempelvis et, der kan beregne, hvordan mennesker og robotter skal bevæge sig for at nå et mål.

Modellering bruges i samfundet

I dag har invers kinematik udviklet sig til mange flere anvendelser, end hvad vi har beskrevet i denne introduktion til emnet. På Datalogisk Institut arbejdes der i skrivende stund med at kombinere invers kinematik-modeller med statistik i et forsøg på at gøre det muligt at lære en computer at kunne se og afkode ved hjælp af et kamera, hvordan mennesker bevæger sig. På denne måde kan vi konstruere nye og bedre måleapparater, som andre forskere igen kan bruge til at opnå endnu bedre forståelse af, hvordan vi mennesker virker.

Der findes mange andre eksempler på anvendelser. Man kunne fx lære en robot, hvordan den skal løse en opgave ved at lade den se og kopiere, hvordan et menneske foretager en eller anden bevægelse, eller man kunne konstruere en computertræner, der kan lindre ældre menneskers gigt gennem fysioterapi eller træne yngre mennesker, der har fået sportsskader. Disse samfundsgavnlige anvendelser af computerens mange muligheder skaber en meningsfyldt hverdag for os forskere på Datalogisk Institut, der arbejder sammen med læger, fysioterapeuter og idrætslærere, samtidig med at vi bruger masser af matematik og koder programmer, der hjælper andre mennesker.

Nedenfor er en oversigt over de matematiske emner, vi har brugt i modelleringsprocessen:

- Homogene koordinater
- Vektorfunktioner
- Differentiering af vektorfunktioner
- Ikke-lineær optimering
- Steepest Descent metoden. ❖

Læs mere

Robert Messer: Linear Algebra, Gateway to Mathematics, HarperCollins College Publishers

Robert A. Adams: Calculus, A complete course, Addison-Wesley

Jorge Nocedal og Stephen J. Wright: Numerical Optimization, Springer

Kenny Erleben



Kenny Erleben har været lektor på DIKU siden 2009 og forsker bl.a. i fysisk baseret simulation og anvendt matematik. Han blev kandidat i datalogi i 2001 og ph.d. i 2005. Kenny er leder af eScience-forskerskolen ved Det Naturvidenskabelige Fakultet, har været formand

for OpenTissue-open source-projektet siden 2007 og modtog et NVidia-professor-partnerskab i 2008.

Kenny underviser bl.a. i computergrafik, numerisk optimering og computational physics. Af udvalgte artikler kan nævnes: "Physics-Based Animation" (2005) og "Velocity-Based Shock Propagation for Multibody Dynamics Animation" (2007).

Optimering - et grundlag for bæredygtig it

.....
Af **David Pisinger**, adjungeret professor, DIKU
.....

Da bogen "Grænser for vækst" (Meadows m.fl.) udkom i 1972, rystede den store dele af samfundet. Verden havde ellers set lidt af hvert: Vietnamkrigen, Watergateskandalen og den kolde krig. Mange mente, at nu kunne det ikke blive værre. Men alligevel fik bogens dommedagsprofeti mange rynkede øjenbryn frem.

Bogen forudsagde, at hvis befolkningsekspllosionen fortsatte som hidtil, ville jordens ressourcer snart være opbrugt. Der ville ske en voldsom økonomisk afmatning, og livskvaliteten ville falde i starten af 2000-tallet. Resultaterne var baseret på en global computermodel, "World3", udviklet på MIT, der gennem komplekse ligningssystemer gjorde det muligt at simulere klodens tilstand mange år frem i tiden. Kort efter udgivelsen kom oliekrisen som en slags bekræftelse af dommedagsprofetierne.

Bogen var revolutionerende i flere henseender:

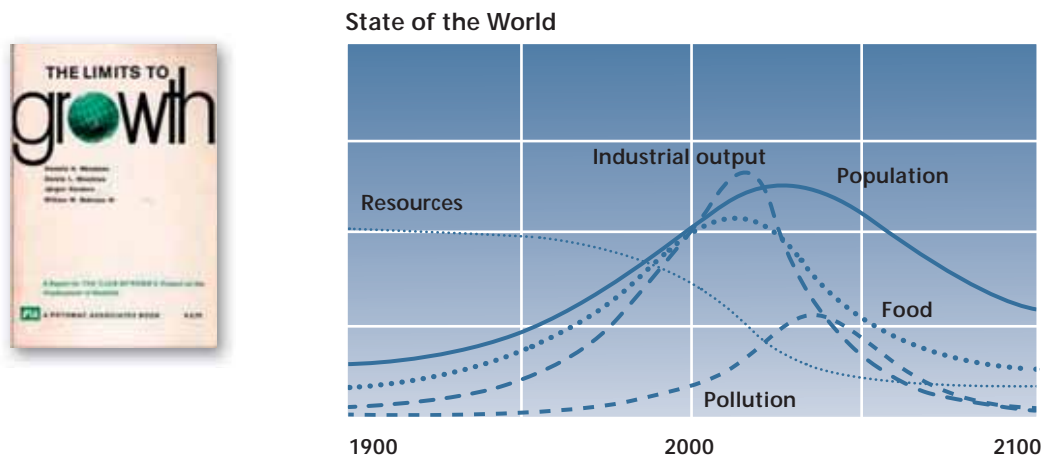
1. Det var den første populære anvendelse af it-baseret simulation af sammenhænge mellem livets forudsætninger. Modellen beskrev gensidige afhængigheder og afspejlede derfor i højere grad den virkelige verden.
2. Der blev tegnet et dystert billede af fremtiden. Hvis befolkningsekspllosionen fortsatte, og vi blev ved med at ødsle med jordens ressourcer, ville vi snart se katastrofale følger.
3. Bogen stillede spørgsmålstejn ved den bibelske opfattelse af, at mennesket hersker over jorden. Mennesket er ligesom alle andre levende væsner underlagt nogle spilleregler på kloden.

Her, knap 40 år efter udgivelsen, er "Grænser for vækst" mere aktuel end nogensinde. Vi kæmper stadig med en eksplosivt voksende befolkning, men hvor bogen oprindeligt var bekymret for, at mængden af fossilt brændstof var begrænset, frygter vi nu i højere grad konsekvenserne af CO₂-udslip, drivhusgasser og forurening.

It som vagthund og frelsende engel

På samme måde som it blev brugt til at forudsige grænserne for den globale vækst, kan it også være vejen til at løse problemerne. Eller i hvert fald en del af løsningen. Med en voksende befolkning og færre råstoffer må vi blive bedre til at udnytte vores ressourcer, hvis vi vil opretholde vores nuværende levestandard.

Figur 1: Bogen "Grænser for vækst" satte med sine computerbaserede simulationer fokus på bæredygtighed. Som det ses på figuren til højre, vil industri- og fødevareproduktionen aftage drastisk omkring 2020, og forureningen vokse frem til 2040, hvor befolkningstallet daler voldsomt.



Der bliver i disse år brugt mange penge på at investere i ny teknologi, som kan mindske energiforbrug og begrænse CO₂-udslip, men omstillingsprocessen er langsom og dyr, og visse teknologiske løsninger fører blot nye problemer med sig. Tænk blot på a-kraft-værker, enorme vindmølleparker i naturen osv. Det kan derfor virke besnærende i stedet at blive bedre til at udnytte de eksisterende ressourcer ved brug af it-baseret optimering.

Det startede ellers på en mindre flatterende måde. Første gang man begyndte at bruge kvantitative metoder til beslutningstagning, var under anden verdenskrig, hvor faget *operationsanalyse* så dagens lys. Operationsanalyse bruger regnekraft til at støtte beslutninger og opererer derfor i grænseområdet mellem matematik og datalogi. Under anden verdenskrig var målet at få ødelagt flest mulige af fjendens fly og skibe, og mere end 1.000 medarbejdere blev brugt til at planlægge krigens forløb i England. Efter krigens afslutning så man mulighederne i at bruge de samme teknikker inden for planlægning af fx produktion, logistik og infrastruktur. Især fremkomsten af lineær programmering i 1947 satte skub i tingene, og det viste sig hurtigt, at der næsten ikke var grænser for, hvad metoderne kunne bruges til. Fokus har dog næsten altid været på at maksimere fortjenesten (i en eller anden form) af sine handlinger. I dag står vi nu ved en vigtig korsvej, hvor interessen mere og mere drejer sig om at finde bæredygtige løsninger.

Ud fra et datalogisk synspunkt er optimeringsproblemer interessante at beskæftige sig med. Der findes ikke nogen universel løsningsmetode, som kan løse alle optimeringsproblemer, og teorien om NP-fuldstændighed (se faktaboks) giver os god grund til at tro, at vi aldrig vil blive i stand til at løse disse rutinemæssigt. Faktisk er der udlovet en million dollar i præmie til den, der kan løse et af disse problemer effektivt.

Lette og svære optimeringsproblemer

Forestil dig, at du som rektor på et gymnasium skal dele 100 studerende i tre klasser. Alle studerende har skrevet en liste over personer, de ikke vil i klasse med, og vi søger en opdeling, som respekterer alle disse ønsker. Problemet er et eksempel på, hvad forskere kalder et NP problem, idet det er nemt at kontrollere, om en foreslået opdeling af de studerende overholder alle ønsker. Men at finde en opdeling, som overholder alle ønsker, er en uoverkommelig opgave. Hvis man bare prøver sig frem med alle mulige opdelinger, er der mere end 3^{100} kombinationsmuligheder, et tal, der overstiger alle atomer i universet! Så hverken vi eller fremtidige civilisationer vil kunne bygge en supercomputer, som vil kunne prøve alle kombinationsmuligheder. Men måske findes der en snedig algoritme, som kan løse problemet uden at prøve alle kombinationsmuligheder. Vi har bare endnu ikke udviklet matematiske metoder til at gøre dette.

Et af datalogiens store spørgsmål er, om der findes spørgsmål, hvor det er nemt at kontrollere, om en løsning er korrekt, men som kræver ufattelig lang tid at løse ved direkte beregning. Problemet ovenfor ligner et sådant problem, men hidtil har ingen været i stand til at vise, at man rent faktisk skal bruge ufat-

telig lang tid på at løse det. Stephen Cook og Leonid Levin formulerede dette spørgsmål i 1971 som P (det er nemt at løse problemet) versus NP (det er nemt at kontrollere en løsning) -problemstillingen.

Hvis det en dag bliver vist, at $P = NP$, vil det få omfattende konsekvenser for vores tilværelse og tænkemåde. Man vil fx kunne få en computer til at bevise alle uløste påstande inden for matematik og fysik. Det er nemlig nemt at kontrollere, om et bevis er korrekt, så problemet "find et matematisk bevis" tilhører klassen NP. Hvis $P = NP$, vil det også være nemt at "finde et matematisk bevis". Hvis man kunne få en computer til at bevise uløste matematiske problemer, ville man kunne tale om ægte kunstig intelligens, og teknisk forskning ville rykke et kvantespring fremad.

Blandt NP-problemerne er der nogle særlig svære problemer, som vi kalder NP-fuldstændige problemer. Disse problemer er sværere end alle NP-problemer, i den forstand at man kan transformere alle NP-problemer til et NP-fuldstændigt problem.

Nedenstående tabel viser nogle lette (P-problemer) og svære problemer (NP-fuldstændige problemer)

Korteste vej	Givet et vejnet, find den korteste vej fra a til b	Let
Længste vej	Givet et vejnet, find den længste kredsfrø vej fra a til b	Svært
Todeling	Givet et antal heltal, del disse i to mængder, så summen af tallene er ens i hver mængde	Svært
Handelsrejsendes problem	Givet et vejnet, find den korteste tur rundt, som besøger alle byer	Svært
Primtal	Givet et heltal, afgør, om tallet er et primtal	Let
Deling i to klasser	Givet et antal studerende, hvor hver studerende har angivet, hvem han/hun ikke vil i klasse med. Del de studerende i to klasser så ønskerne overholdes	Let
Deling i tre klasser	Givet et antal studerende, hvor hver studerende har angivet, hvem han/hun ikke vil i klasse med. Del de studerende i tre klasser, så ønskerne overholdes	Svært

Hvert enkelt optimeringsproblem er derfor en videnskabelig udfordring, som kræver et samspil af mange discipliner. Fra matematik over algoritmik til design af supercomputere. Mange optimeringsproblemer er særdeles beregningstunge at løse, så grid og cloud computing er velkomne værktøjer til at fordele beregningerne på mange computere. Men selv hvis vi havde alle jordens computere til rådighed, ville der være visse optimeringsproblemer, som stadig ikke ville kunne blive løst (se figur 2). I stedet er vi nødt til at udvikle nye matematiske løsningsmetoder for at kunne hamle op med de sværeste optimeringsproblemer, og selv her må vi undertiden erkende, at vi ikke kan løse et problem (endnu).

<i>n</i>	10	50	100	300	1.000
5 <i>n</i>	50	250	100	1.500	5.000
<i>n</i> ²	100	2.500	10.000	90.000	10 ⁷
<i>n</i> ³	1.000	125.000	10 ⁷	10 ⁸	10 ¹⁰
2 ^{<i>n</i>}	1.024	10 ¹⁶	10 ³¹	10 ⁹¹	10 ³⁰²
<i>n</i> !	10 ⁷	10 ⁶⁵	10 ¹⁶¹	10 ⁶²³	"stort"

Figur 2: Mange optimeringsproblemer tager eksponentielt lang tid at løse, fx 2^{*n*}, hvor *n* er antallet af variable, der skal bestemmes. Problemerne behøver ikke at blive ret store, førend vi må opgive at løse dem. Tabellen viser, hvor hurtigt 2^{*n*} vokser i forhold til andre funktioner. Til sammenligning er der gået ca. 13,73 milliarder år siden Big Bang, hvilket er omkring 10¹⁸ sekunder. Hvis en computer havde kørt uafbrudt siden Big Bang og udført en milliard regneoperationer i sekundet, ville den nu have udført 10²⁷ regneoperationer. Så et problem med *n* = 100 variable, hvor det tager 2^{*n*} regneoperationer at løse det, ville være kommet gennem mindre end 1 promille af beregningerne.

Produktion og bæredygtighed som ligeværdige mål

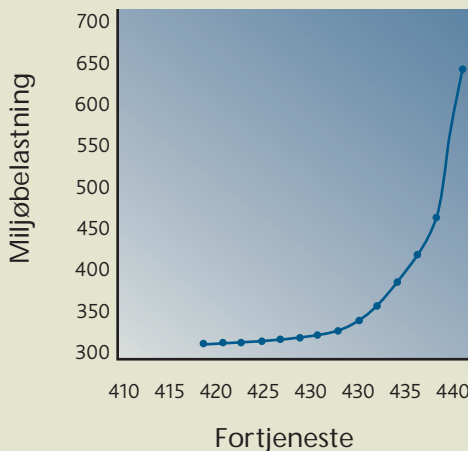
Det nytter ikke noget, at man alene søger efter de mest bæredygtige løsninger, for her vil de matematiske modeller typisk svare, at det miljømæssigt bedste er ikke at producere eller transportere noget som helst. Selvom dette er en smuk løsning, er det de færreste ledere, som ville blive genvalgt med et sådant valgprogram.

Det er derfor vigtigt, at vi ser produktion, fortjeneste og bæredygtighed som ligeværdige mål. Når fortjenesten presses til det yderste, ser man ofte, at ressourceforbrug og forurening vokser dramatisk. Men blot en lille formindskelse af fortjenesten på 1-2 procent kan ofte give en voldsom formindskelse af ressourceforbruget. Dette hænger sammen med, at der ofte kun er én løsning, som maksimerer fortjenesten, mens der findes rigtig mange løsninger, som er bare nogle få procent dårligere. Blandt disse mange løsninger er det derfor relativt nemt at finde en løsning, som også er bæredygtig.

Når man går fra at maksimere fortjenesten til at afveje profit mod bæredygtighed, er der ikke en entydig optimal løsning. Derimod findes der en vifte af løsninger som illustreret på figur 3. Man kan ikke uden videre bruge en computer til at vælge en af disse løsninger, idet de alle er lige gode på hver sin måde. Man må derfor i stedet lade en beslutningstager afveje de forskellige kriterier mod hinanden. I sidste instans er det en politisk-strategisk afvejning, hvor grænsen mellem produktion og bæredygtighed skal ligge, men vi kan bruge computeren til at finde de bedste valgmuligheder og synliggøre konsekvenserne af hvert valg.

En øget miljøbevågenhed kan også have mange implicitte fordele. For at spare energi vil man ofte vælge at sejle/køre lidt langsommere end den maksimale hastighed. Sådanne løsninger fører til mere robuste planer, idet det er nemmere at indhente forsinkelser. I sidste instans fører det til større kundetilfredshed og at varer leveres til tiden.

I de følgende afsnit gennemgås nogle projekter, hvor DIKU har bidraget til forskningsprojekter, der munder ud i større bæredygtighed.



Figur 3: Miljøbelastning som funktion af fortjenesten.



Det kan give 4-5 procent besparelse på transportudgifterne at fylde containerne bedre

Kunsten at pakke sofaer i containere

I 2004 påbegyndte forskere på DIKU et samarbejde mellem Bari Universitet og en førende italiensk sofaproducent (Natuzzi SpA) omkring udvikling af it-baserede metoder til pakning af sofaer i containere (Egeblad m.fl.). Der bliver årligt transporteret flere hundrede tusind containere med møbler rundt omkring i verden, så en reduktion af transportomkostningerne på blot 4-5 procent ville betyde en stor gevinst for miljøet og for virksomheden.

Hvad der virker som en intuitivt let opgave, viste sig hurtigt at være en overordentlig svær problemstilling, idet sofaer som bekendt ikke er rektangulære. Da opgaven blev påbegyndt, fandtes der kun nogle få algoritmer til pakning af ikke-rektangulære emner, og disse kunne kun håndtere problemer med 10-20 emner. En container med møbler indeholder ofte mere end 100-150 emner, så der var behov for at udvikle helt nye løsningsmetoder.

Undervejs i denne proces blev Minkowskis berømte arbejde fra 1900-tallets begyndelse taget op. Minkowski opstillede en matematisk formel for, hvordan et legeme kan bevæge sig omkring et andet legeme ude at støde ind i det, og denne viden kan bruges til at beregne, hvor to sofaer kan placeres i forhold til hinanden.



Figur 4: Pakning af sofaer i containere. Med programmet til højre kan man finde en optimal pakning. Dermed er det blevet muligt at reducere transport omkostningerne med 4-5%. Den svævende sofa på figuren til højre markerer at der ikke var plads til denne enhed i containeren.

Alligevel er antallet af mulige måder at pakke 100 sofaer på astronomisk, så man er nødt til at begrænse søgningen til nogle fornuftige kombinationer. Dette kan fx være, at to ens sofaer bliver parret sammen med den indvendige foring mod hinanden, således at de beskytter hinanden under transport. En anden kombination kan være at anbringe en let stol oven på en tung sovesofa.

For at kunne pakke sofaer er det nødvendigt at opstille en generel repræsentation af, hvordan en sådan sofa ser ud, ud fra nogle givne mål. Værktøjet kan anvendes baglæns, således at man faktisk ved at indtaste nogle mål kan designe sin egen sofa, og er et eksempel på, hvordan en god matematisk forståelse kan åbne op for helt nye anvendelsesområder.

Det udviklede program bliver nu brugt på alle produktionssteder, og det kan give 4-5 procent besparelse på transportudgifterne ved at fylde containerne bedre. Samtidig undgår man transportkader, idet lasten ikke forrykker sig så let, når containeren er fyldt helt op.

Fra et forskningsmæssigt synspunkt har det ført til den første 3-dimensionelle pakningsalgoritme, som kan løse praktiske problemer fra det virkelige liv, hvor emnerne ikke er begrænset til at være rektangulære. Dette har revolutioneret produktionsprocessen, idet man nu kan simulere pakning af en container alle steder i virksomheden. Således kan man på forhånd teste, om nogle bestilte varer kan være i en enkelt container. Hvis programmet kan pakke alle sofaerne, så der stadig er plads i containeren, vil forhandleren typisk bestille endnu flere varer for at få fyldt containeren. Når transportudgifterne er betalt, kan man lige så godt udnytte pladsen. Dette fører til et øget salg og i sidste instans også til lavere produktionsomkostninger.

Energieffektiv transportplanlægning

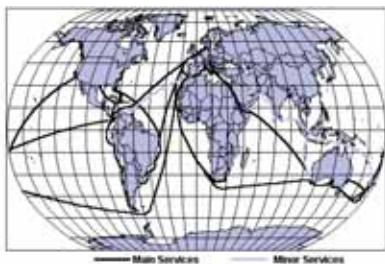
EU har sat sig som mål at reducere CO₂-udledningen med 20 procent inden for det næste årti. Transportsektoren er ansvarlig for en stor del af denne udledning grundet sin afhængighed af fossilt brændstof. Transportsektorens natur gør det svært at finde alternative energikilder, hvorfor den vigtigste metode p.t. kan være at udnytte energien bedre gennem forbedret logistik.

ENERPLAN-projektet (Energieffektiv Transportplanlægning) har til formål at udvikle intelligente it-baserede værktøjer til planlægning af containerbaseret transport. Især skal der udvikles beslutningsstøtte til at planlægge rutenet, så en bedre balance mellem skibenes kapacitet og lastbehov opnås. Endvidere udvikles algoritmer til at laste skibe hurtigere og bedre, således at tiden i en havn minimeres. Dermed kan skibet sejle langsommere på de mellemliggende strækninger og reducere energiforbruget. Endelig er det målet at udvikle planlægningsværktøjer i stil med "Rejseplanen" for containertransport, der viser både pris, rejsetid og miljøbelastning. På den måde kan man vælge den billigste og mest miljøvenlige transportform (se faktaboks).

Projektets mål er at reducere energiforbruget i liner shipping med 3-5 procent ved at designe mere effektive rutenet og ved at forbedre den logistiske håndtering af containerne. For en stor liner shipping-virksomhed svarer en besvarelse på 3-5 procent til CO₂-udledningen fra en større dansk provinsby.

Forskningen er et samarbejde mellem DTU/DIKU, IT-Universitetet og Maersk Line. Ved at have verdens største containershipping-virksomhed med i projektet sikres der adgang til nyeste data om containertransport, samt at de udviklede metoder hurtigt kan omsættes til praksis og dermed spare CO₂-udledning.

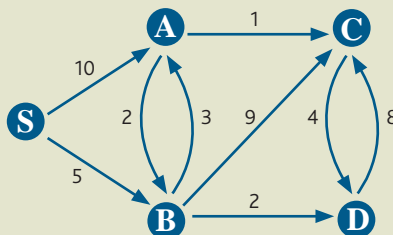




Figur 5: Ved at beregne bedre rutenet for containerskibe er det målet at spare 3-5 % af energiforbruget. For en stor shippingvirksomhed svarer dette til energiforbruget i en større dansk provinsby.

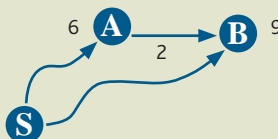
Hvordan finder man den mest miljørigtige transportvej?

Antag, at vi vil sende en container fra Singapore (*S*) til Copenhagen (*C*) på den mest miljørigtige måde. Der er givet en sejlplan for alle containerskibe, hvor knuderne (cirklerne) angiver havne, og kanterne (stregene) angiver sejlruiter. For hver sejlroute er det oplyst, hvor stort et energiforbruget er pr. container. Vi vil kalde dette tal kantens længde eller afstand.



På ovenstående figur er afstanden 2 fra *A* til *B*, mens afstanden fra *B* til *A* er 3 (idet skibet sejler en anden vej). Hvis vi vil finde den mest miljørigtige vej fra byen *S* til byen *C*, kan vi enten gå $S \rightarrow A \rightarrow C$ (længde 11), $S \rightarrow A \rightarrow B \rightarrow C$ (længde 21), $S \rightarrow A \rightarrow B \rightarrow D \rightarrow C$ (længde 22), $S \rightarrow B \rightarrow A \rightarrow C$ (længde 9), $S \rightarrow B \rightarrow C$ (længde 14) eller $S \rightarrow B \rightarrow D \rightarrow C$ (længde 15).

For at løse problemet vil vi på hver knude skrive et tal a , som angiver den for tiden bedst kendte afstand fra *S* til knuden. Til at begynde med er afstanden uendelig for alle knuder, idet vi ikke kender nogen vej til dem endnu. Undervejs i løsningsprocessen vil vi formindske værdierne af a , efterhånden som vi finder *genveje*. På nedenstående figur er den korteste kendte afstand fra *S* til *A* sat til $a = 6$, mens $a = 9$ for knuden *B*. Nu betragter vi kanten $A \rightarrow B$. Da afstanden til *A* plus længden af kanten $A \rightarrow B$ er $6 + 2$, har vi fundet en genvej til *B*, således at vi kan rette den korteste kendte afstand fra *S* til *B* til værdien $a = 8$.

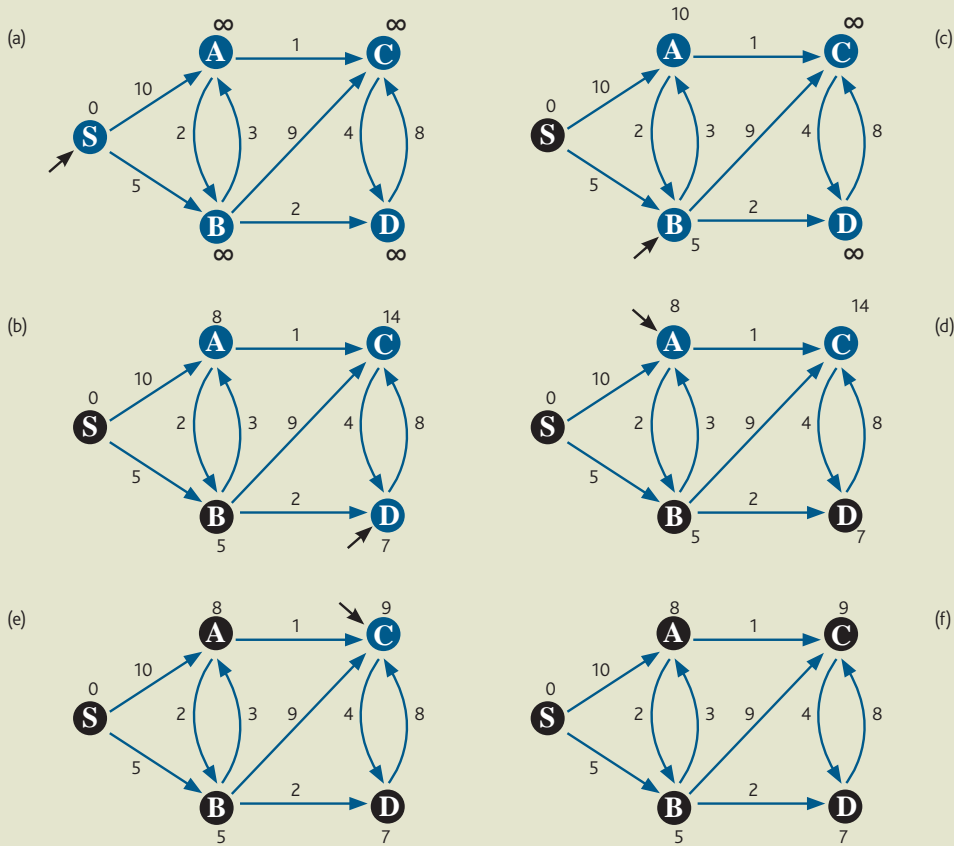


Hvis vi finder *genveje* på en systematisk måde, kan vi beregne afstanden fra S til alle andre knuder med **Dijkstras algoritme**.

1. Lad a være uendelig for alle knuder, og farv alle knuder hvide.
2. Lad a være 0 for startbyen S .
3. Gentag skridt 4 til 6, så længe der er hvide knuder.
4. Vælg den hvide knude, som har lavest værdi af a , kald den x .
5. For alle kanter $x \rightarrow y$, undersøg, om der er en *genvej* fra x til y .
6. Farv knuden x sort.

Eksempel: På nedenstående figur (a) er værdien af a sat til ∞ for alle knuder undtagen S , der har afstanden 0. Den hvide knude, som har lavest værdi af a , er knuden S . Vi undersøger nu, om der er en *genvej* via kanten $S \rightarrow A$ (her bliver afstanden $0 + 10$) og kanten $S \rightarrow B$ (her bliver afstanden $0 + 5$). Da de tidligere værdier af a var uendelig for A og B , er begge kanter en *genvej*, så vi kan opdatere værdien af a på begge knuder. Knude S farves sort.

På næste figur (b) er det B , som er den hvide knude, der har lavest værdi af a . Vi undersøger *genveje* via kanterne $B \rightarrow A$ (afstand $5 + 3$), $B \rightarrow C$ (afstand $5 + 9$) og $B \rightarrow D$ (afstand $5 + 2$). Alle er *genveje*, så værdierne af a opdateres. Knude B farves sort. I figur (c) finder vi, at D er den hvide knude med lavest værdi af a . Der er kun én kant, $D \rightarrow C$, hvor vi skal kontrollere for *genveje*. Her finder vi afstanden $7 + 8 = 15$, men denne er *ikke* en *genvej*, så vi beholder værdien af $a = 14$ i knude C . Knude D farves sort. Således fortsættes indtil sidste figur (f), hvor alle knuder er sorte, og for hver knude angiver a længden af den korteste vej fra S til knuden.



Bedre logistik kan være vejen til klimamål

Den nuværende debat om klimaændringer er for en stor dels vedkommende fokuseret på teknologiske løsninger som elektriske biler, vindkraft osv. Men sådanne løsninger er dyre og tidskrævende at implementere. It-baserede logistik- og planlægningsværktøjer med fokus på miljøet kan være et bedre alternativ, da de ikke kræver dyre investeringer. De er hurtige at implementere, og de underliggende teknikker er modne.

Selvom bæredygtig it ikke i sig selv løser alle miljøproblemer, er det et vigtigt supplement til de mere teknologiske løsninger, for at vi kan nå vores klimamål i tide.

Hvis vi kigger 40 år frem i tiden, vil man med bedre optimeringsmetoder kunne undgå meget spild af ressourcer i produktion og samfund: En stor del af verdens lastbiler kører lige nu tomme eller halvtomme rundt. Der produceres en masse varer, som aldrig bliver solgt eller brugt. Vi bruger alt for mange materialer til at bygge broer og huse, fordi vi ikke kan beregne den samlede bærestyrke. Hjemmehjælperne bruger en stor del af deres dag på at køre rundt mellem klienterne. Der er en mængde problemer at tage fat på, og i takt med at løsningsteknikkerne bliver bedre, vil man hele tiden finde nye anvendelsesområder.

Man kunne forestille sig, at optimering om 40 år blev en integreret del af internettet, således at man kunne se på resourceforbrug som en samlet helhed. Dermed ville man undgå meget spildproduktion, fordi man straks kunne reagere på ændrede behov. Omvendt åbner det op for nogle etiske overvejelser, idet vi måske ikke vil dele viden om vores forbrugsvaner med andre. ♦

Læs mere

Donella H. Meadows, Dennis L. Meadows, Jørgen Randers, William W. Behrens III.

The Limits to growth, Universe Books (1972).

David Pisinger "ENERPLAN - Green Logistic Solutions". *ERCIM news*, 79 (2009).

Clay Math Institute, "Millennium Prize Problems" <http://www.claymath.org/millennium/> (2000).

Jens Egeblad, Claudio Garavelli, Stefano Lisi, David Pisinger "Heuristic for container loading of furnitures" *European Journal of Operational Research* 200, 881-892, (2009).

Stefan Ropke, David Pisinger "A general heuristic for vehicle routing problems" *Computers & Operations Research*, 34, 2403-2435 (2007)

DAVID PISINGER



David Pisinger har været professor i algoritmik og optimering på Datalogisk Institut ved Københavns Universitet siden 1993 og har siden 2009 været professor i operationsanalyse på DTU Management. David blev bachelor i matematik i 1988 og kandidat i datalogi i 1990. Han fik ph.d.-graden i datalogi i 1995

og modtog samme år Danmarks Naturvidenskabelige Akademis ph.d.-pris for sin afhandling.

Gennem årene har han været medlem af programrådet for flere ledende konferencer inden for operationsana-

lyse (bl.a. IFORS 2011, EURO 2000 og EURO 2010) samt bidraget til at arrangere ISMP 2003, et internationalt symposium om matematisk programmering i Danmark. Han har ledet flere større forskningsprojekter, bl.a. i samarbejde med Bari Universitet om bedre udnyttelse af containere og med Maersk Line om energibesparende transportveje.

David underviser bl.a. i algoritmik og operationsanalyse og har fungeret som vejleder i adskillige specialer og ph.d.-projekter. Blandt udvalgte publikationer kan nævnes bogen "Knapsack Problems" fra 2004 og artiklerne "Heuristics for Container Loading of Furniture" (2010) samt "A General Heuristic for Vehicle Routing Problems" (2007).

Machine learning – automatisk læring med computere

.....
Af **Mads Nielsen**, DIKU
.....

Begrebet *machine learning*, som på dansk vel mest passende kan gengives med ”automatisk læring”, er baseret på det særlige område inden for datalogisk forskning, der hedder *algoritmer* kombineret med statistisk analyse. Det går ud på at lade computeren lære at genkende mønstre ud fra eksempler og data og vel at mærke lære mere end eksemplerne selv: På basis af data kan computeren også lære at generalisere til nye eksempler og træffe såkaldt intelligente beslutninger.

Et illustrerende eksempel

I 1994 var jeg på vej med bussen til DIKU til en ph.d.-skole, som handlede om læring ved hjælp af *minimum description length-princippet*, sad en vel 5-årig dreng på sædet foran mig i bussen. Han sagde til sin mormor: ”Se, der er en bus med flag på!” Et øjeblik efter kom næste bus: ”Se, mormor, alle busser har flag på i dag!” Han generaliserede ud fra to eksempler til en regel. Gennem *machine learning* forsøger computeren at gøre det samme: at finde den hypotese, der passer bedst til de erfarede data og samtidig har størst mulig chance for at passe til fremtidige data.

Med udviklingen af computeres lager- og beregningskapacitet og også som følge af metodernes udvikling over de sidste årtier er machine learning blevet et konkurrencedygtigt alternativ til at opfinde den dybe tallerken selv. Som eksempler på services, der anvender machine learning, kan nævnes Googles søgemaskine, stemmegenkendelse i telefonsvarere, kunstige modspillere i computerspil, forudsigelse af aktiekurser og programmer til visuel genkendelse af ansigter (prø fx Polar Rose www.polarrose.com, der automatisk genkender personer i dine fotografier), men også mere prosaiske eksempler som centrifugeringsprogrammer i vaskemaskiner. Senest har Volvo kørt rundt i København for at arbejde med deres machine learning-algoritme til automatisk genkendelse af fodgængere i bybilledet. Teknikken er lanceret i 2010 med det formål at få bilen til at bremse automatisk for at undgå kollision.

Kunstig intelligens vs. machine learning

Kunstig intelligens er et begreb, der længe har spøgt i litteraturen, og som offentligheden har stiftet bekendtskab med i bl.a. Arthur C. Clarkes *Rumrejsen år 2001* fra 1968 (filmatiseret af Stanley Kubrick) og Brian Aldiss’ *AI* fra 2001 filmatiseret af Steven Spielberg. Machine learning kan opfattes som en del af dette begreb. I faglitteraturen er begreberne dog typisk adskilte: Hvor kunstig intelligens i højere grad beskæftiger sig med at finde løsninger til komplekse, men fuldt ud specificerede problemer, beskæftiger machine learning sig med at udlede regler fra data. Et godt eksempel er de populære sudokuer, hvor man med brug af kunstig intelligens ville indkode de kendte regler og lave et program, der finder en løsning til at udfylde spillepladen med tal.

Machine learning-tilgangen kræver ikke, at man kender reglerne for sudoku, men derimod at man ser nogle løsningseksempler. Computeren vil derefter forsøge at regne reglerne ud og samtidig lave et program, der kan anvende reglerne til at finde nye løsninger.

Denne tilgang kaldes også *inferens* eller *ræsonnement*. Kunstig intelligens anvender *deduktiv inferens*, dvs. slutter sig frem til løsningen ved brug af udelukkelsesmetoden (Sherlock Holmes-metoden), mens machine learning benytter *induktiv inferens*, dvs. en form for sandsynliggørelse gennem sammenligning med lignende eksempler. Sidstnævnte er, hvad matematikerne kalder et ”dårligt stillet” problem (Hadamard) – et problem uden en entydig løsning. Drengen i bussen ville kunne finde mange forskellige regler for, hvilke busser der har flag, og hvilke der ikke har flag. Vel at mærke regler, der alle er i overensstemmelse med hans observationer af, at de to første busser har flag. Han valgte reglen ”Alle busser har flag på i dag”, men kunne lige vel have valgt reglen ”Alle busser på Christianshavns Torv har flag på” eller for den sags skyld en mere kompliceret regel som ”Når jeg tæller busser og giver dem fortløbende numre, har alle primtalsbusser et flag på”. Alle tre regler svarer fuldt ud til observationerne, og vi må udvikle et princip til at vælge den regel, der har størst chance for at stemme overens med fremtidige observationer. At vælge dette princip og komme frem til algoritmer på baggrund af princippet er kunsten og videnskaben i machine learning.

Minimum description length-princippet

Fra en lidt mere filosofisk vinkel kan minimum description length (MDL)-princippet nævnes. Det bygger på Occams razor: Den simpleste af lige gode hypoteser er at foretrække. Dette er formaliseret i MDL, hvor man forsøger at indkode data så kompakt som muligt med mange konkurrerende komprimeringsmetoder. Den metode, der så bruger færrest mulige bit til at indkode data, anses som den bedste. Forbindelsen til statistisk machine learning kommer gennem Shannons informationsteori, hvor sandsynlighedsfordelingers entropi er den teoretiske nedre grænse for, hvor mange bit der skal bruges til at indkode instanser trukket fra fordelingen. Der er på denne måde etableret mange forbindelser mellem bayesiansk tankegang og MDL.

Tages MDL-filosofien alvorligt i sin yderste konsekvens, drejer machine learning sig således om at komprimere data mest muligt. Den nederste grænse for komprimeringen kendes også som Kolmogorov-kompleksiteten af data: længden af det korteste computerprogram, der udskriver data. Desværre er det vist, at Kolmogorov-kompleksiteten ikke generelt er beregnelig. Vi kan kun på vores togt efter gode komprimeringer håbe at finde et kort program. Vi kan aldrig garantere, at vi har fundet det korteste. Boris Ryabko har for en stor klasse af data vist, at i gennemsnit vil den ekstra kodelængde, vi får, fordi vi ikke bruger uendelig lang beregningstid, være omvendt proportional med den anvendte beregningstid. Derved har han formaliseret det berømte citat af Blaise Pascal: ”If I had had more time, I would have written you a shorter letter”.

At estimere med statistisk modellering (Bernoulli-processen)

Når vi ønsker at finde den regel, der har størst chance for at generalisere korrekt, bevæger vi os hurtigt ind i sandsynlighedsregning og statistik, når det nu er "chancer", vi taler om. Her vil vi gå gennem Bernoulli-processen som et kanonisk eksempel på mekanismerne i et modelvalg: Vi har en mønt til at slå plat eller krone med, men mønten er muligvis skæv, hvorfor plat og krone ikke giver samme sandsynlighed (se figur 1).

Vi kaster nu mønten gentagne gange og vil forsøge at udregne sandsynligheden for krone i fremtidige kast. Denne proces er det første simple eksempel, men allerede her kan det blive komplekst at finde en god løsning. Den umiddelbare tilgangsvinkel vil være ud fra vores træningskast at udregne den fremtidige sandsynlighed for krone som:

$$\Theta = \text{antal krone} / \text{antal kast} = \text{antal krone} / (\text{antal krone} + \text{antal plat})$$

Dette er maximum likelihood-estimatet af sandsynligheden Θ : Af alle mulige valg af Θ er dette valget, hvor sandsynligheden for det udfald (antal krone og plat), vi har opnået, er størst. Vi beskriver dette som:

$$\Theta_{\text{ml}} = \arg \max_{\Theta} p(D|\Theta)$$

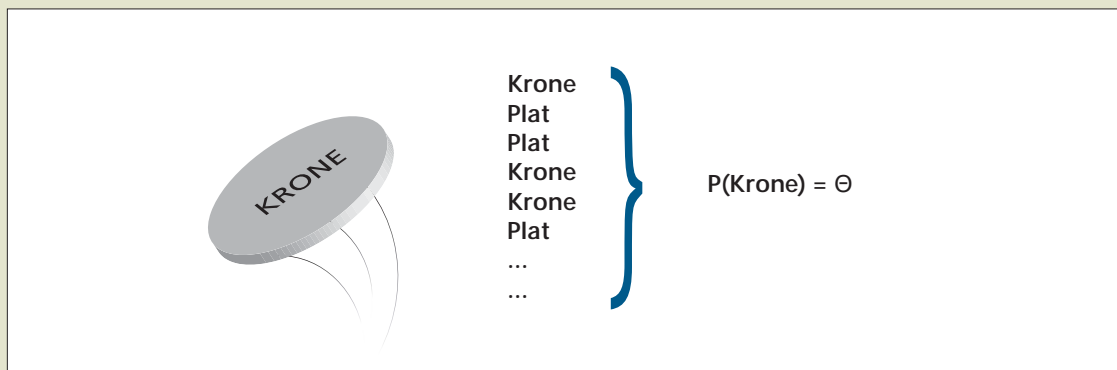
Det vil sige, at *maximum likelihood*-estimatet af Θ (som tilhører intervallet fra og med 0 til og med 1) findes som det Θ , der maksimerer sandsynligheden for data, D (i dette tilfælde antal krone og plat), når værdien af sandsynligheden for krone, Θ , antages kendt. Abstraktionen er altså, at vi gætter løsningen, ser, hvor sandsynlige data er, og så gætter igen og igen, indtil vi har fundet det gæt af løsningen, hvor data er mest sandsynlige.

Maximum likelihood-estimering er standardmetoden for statistisk analyse, men lider i ovennævnte anvendelse af en mangel: Hvis vi kun har kastet mønten en enkelt gang, altså forsøger at lære møntens karakteristika ud fra et enkelt kast, vil Θ blive 0 eller 1, afhængigt af om vi tilfældigvis har fået krone eller plat i første forsøg. Det virker urimeligt, at vi dermed laver en model, som siger, at alle fremtidige kast bliver som det første: Vi bliver ved med at slå krone hver eneste gang. Specielt hvis vi faktisk har haft mønten i hånden og synes, at den minder om en almindelig, fair mønt. Denne forhåndsviden er ikke med i *likelihood-estimatet*.

I stedet for at analysere $p(D|\Theta)$, sandsynligheden for data D , når Θ er kendt, kunne vi analysere $p(\Theta|D)$, sandsynligheden for Θ , når D er kendt. Det virker mere rimeligt at finde sandsynligheden for en model Θ baseret på data. Dette kan gøres ved hjælp af Bayes' formel for a *posteriori*-sandsynligheden for Θ :

$$p(\Theta|D) = p(D|\Theta) p(\Theta) / p(D)$$

Her udregnes sandsynligheden for modellen Θ baseret på likelihoodet $p(D|\Theta)$, forhåndssandsynligheden (eller a *priori*-sandsynligheden) $p(\Theta)$ og evidensen $p(D)$. Evidensen $p(D)$ afhænger ikke af Θ og fungerer som en konstant, der sørger for normalisering af $p(\Theta|D)$, så sandsynlighedsvolumen er 1 summeret over alle værdier af Θ . A *priori*-sandsynligheden $p(\Theta)$ indkoder vores forhåndsantagelser om Θ . Dette er vores væsentligste værktøj, når vi ønsker estimer, der generaliserer bedre end *maximum likelihood*-estimatet.



Figur 1: Bernoulli-processen. En mønt kastes gentagne gange, og vi ønsker at estimere sandsynligheden for udfaldet krone i fremtidige kast.

Ud fra *a posteriori*-fordelingen $p(\Theta|D)$ ønsker vi at vælge et enkelt "bedste" Θ , som vi kalder Θ^* . Dette bedste valg kan foretages efter mange principper, men ofte vil vi vælge den løsning, der minimerer den forventede kvadratfejl.

$$\Theta^* = \arg \min_{\Theta} \int (\Theta - \Theta')^2 p(\Theta'|D) d\Theta' = E(\Theta)$$

Det svarer til *a posteriori*-middelværdien af Θ . Nu kan vi lege med, hvilken *a priori*-fordeling $p(\Theta)$ vi vil antage. Klassisk vil man vælge en ligefordeling på intervallet $[0;1]$. Dette er en såkaldt *ikke-informativ* prior, da vi antager en fordeling, der bestemmer så lidt som muligt. Laplace udregnede under antagelse af ligefordelingen, at:

$$\Theta^* = (\text{antal krone} + 1) / (\text{antal krone} + \text{antal plat} + 2)$$

Løsningen her svarer til, at vi, før vi startede Bernoulli-processen, allerede havde kastet mønten to gange og fået én plat og én krone og derefter benyttede *maximum likelihood*-estimatoren.

Nu da vi har nået målet med et Bayes-estimat, vil vi undersøge dette nærmere: Formlen giver også mulighed for at indsætte resultatet af 0 kast og få $\Theta^* = 1/2$ som løsning. Dette er middelværdien af vores *a priori*-fordeling. Uden data er dette naturligt vores bedste bud.

Man kan ændre forhåndsantagelserne til en anden fordeling. Hvis man antager en beta-fordeling af Θ , kan Bayes-estimatet udregnes som:

$$\Theta^* = (\text{antal krone} + R) / (\text{antal krone} + \text{antal plat} + 2R)$$

hvor R er en parameter i beta-fordelingen. Her kan man som en finurlighed tage grænsen, hvor R sænkes mod 0. I dette tilfælde bliver Bayes-estimatet identisk med *maximum likelihood*-estimatet. Den *a priori*-fordeling, der giver dette Bayes-estimat, er speciel, da det er den fordeling, der betyder, at vi tror helt på, hvad vi opdager i fremtiden. Det er fordelingen, som Jaynes kalder "state of total confusion".

Ovenfor minimerede vi kvadratfejlen og ankom derved til middelværdien som Bayes-estimat. Dette er et fornuftigt men dog vilkårligt valg. Generelt vil man kunne erstatte kvadratfejlen med en anden *tabsfunktion* og derved ændre sin estimator. Et specielt valg er at antage tabsfunktionen, hvor kun det rigtige valg giver lavt tab, og alle andre valg gives et ens og højere tab. For kontinuerte parametre modelleres dette ved minus Dirac-delta-fordelingen. Man får så en Bayes-estimator, der minimerer tabsfunktionen, der er identisk med den, der maksimerer *a posteriori*-fordelingen. Denne kaldes *maksimum-a posteriori*-estimatet, eller normalt i kort form MAP-estimatet.

I kort opsummering: Normalt laves induktiv inferens som *maximum likelihood*-estimat, men Bayes' formel giver mulighed for at tage højde for forhåndsviden og bestemme *a posteriori*-fordelingen. Derudover vil valget af en tabsfunktion gøre det muligt at beregne det forventede tab for alle parameterværdier og vælge den med mindst forventet tab.

Machine learning

Lad os nu kigge en smule mere formelt på, hvad machine learning er. Vi starter processen med et datasæt S med N instanser, som vi skal lære fra, også kaldet træningssættet:

$$S = \{ s_1, s_2, \dots, s_N \}$$

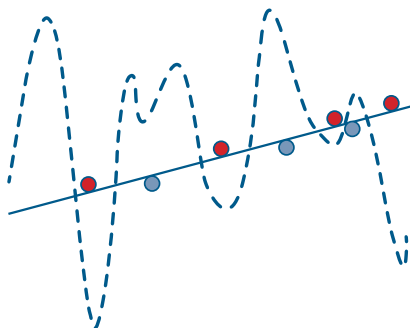
Hver instans består af data (x) med tilhørende label (y : $s_i = (x_i, y_i)$). I eksemplet i tekstboksen med Bernoulli-processen var data, i dette tilfælde nummeret på kastet med mønten, ikke af større betydning, hvorimod udfaldet (label), plat/krone, var vigtigt. I andre tilfælde er data af stor betydning. Et eksempel kan være ansigtsgenkendelse. Her er billedet data (x), og identiteten på personen label (y). Når labels er kendte i træningssættet, taler vi om *supervised learning*. Når labels ikke forefindes, er vi i den vanskeligere situation, der kaldes *unsupervised learning*. Alt, hvad vi kan gøre her, er at finde naturlige grupperinger af data og selv opfinde en label til disse grupperinger. I *supervised learning* forsøger vi at finde en model, der afbilder data til labels: $m: x \rightarrow y$. Modellen skal gerne afbilde så korrekt som muligt både for træningssættet og for hidtil usete instanser, så den også generaliserer godt. Modellen vælges som én blandt mange modeller i hypoteserummet $H = \{m_1, m_2, m_3, \dots\}$. Ligesom vi i forbindelse med Bernoulli-processen valgte *maximum likelihood*, kan vi her vælge den model, der har det laveste antal fejl på træningssættet. Denne strategi kalder vi *empirisk fejlminimering*. Vi kan nu også vælge en tabsfunktion, hvor forskellige fejltypen vægtes forskelligt. Med tabsfunktionen inkluderet kalder vi det for *empirisk risikominimering*.

Machine learning-algoritmer, der implementerer empirisk risikominimering, har mange gode egenskaber. Man kan vise, at man finder den bedste model i hypoteserummet, hvis træningssættet er stort nok. I dette tilfælde siger man, at algoritmen er konsistent. Endvidere kan man vise, at algoritmen generaliserer. Generalisering er den vigtige egenskab til sikring af, at hidtil usete data også får rette label, så godt det nu kan lade sig gøre i hypoteserummet. At minimere risikoen på træningssættet til en meget lille risiko betyder blot, at man har en meget fleksibel modelklasse i sit hypoteserum. At opnå en meget lav risiko på hidtil usete data, dvs. data, som ikke er indgået i at udvælge modellen i hypoteserummet, er gavnligt, så algoritmen faktisk kan rubricere nye data. Med empirisk risikominimering generaliserer man i matematisk forstand, men desværre kun når træningssættet bliver stort nok. Ved Bernoulli-processen klarer *maximum likelihood*-estimatet sig også godt, når der er data nok at lære fra. Derimod klarer det sig dårligere på usete data, hvis det kun har ganske få kast at lære fra. Derfor erstattede vi *maximum likelihood*-estimatet med Bayes-estimatet. Den tilsvarende regel ved små træningssæt benyttes også i machine learning.

Kunsten er ud fra et begrænset træningssæt at opnå så god generalisering som muligt

Hvis hypoteserummet er meget stort, kan man potentielt finde en hypotese, der generaliserer

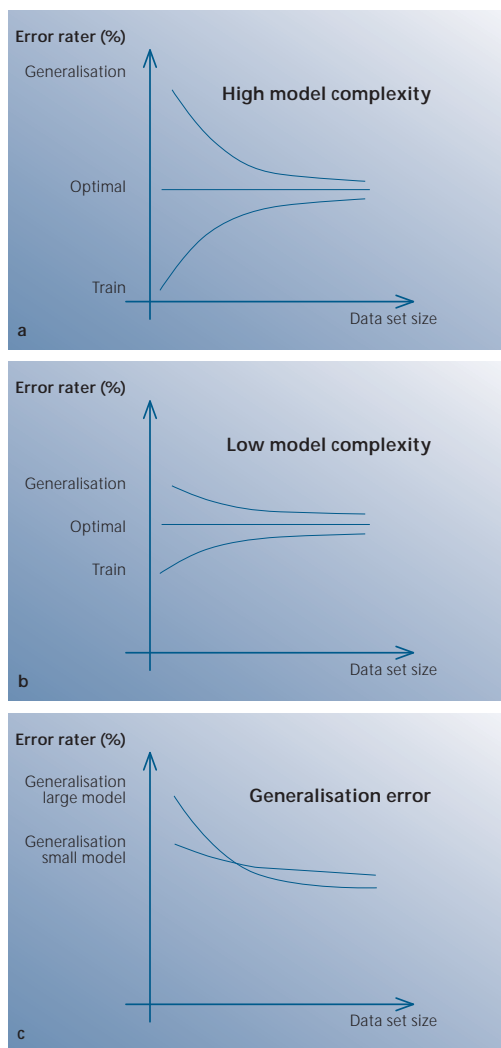
meget godt. Desværre vil der også være mange hypoteser, der tilfældigvis har en lille fejl på træningssættet, men ikke generaliserer godt (se figur 2).



Figur 2: Består hypoteserummet kun af lige linjer, kan man adskille de røde og blå bolde på hver sin side af linjen. Hvis man vælger et hypoteserum med meget større kapacitet, kan man stadig adskille blå og røde bolde, endda med større margin. Spørgsmålet er blot, hvad der sker, når nye bolde dukker op. Det er svært at tro, at den bugtede kurve tilfældigvis vil generalisere vel.

Generelt taler man om hypoteserummets kapacitet. Jo større hypoteserum med mere fleksibel modelklasse, jo større kapacitet. Når kapaciteten bliver større, vil træningsfejlen altid blive mindre for fornuftige algoritmer. Derimod vil generaliseringsfejlen ikke nødvendigvis blive mindre, da den model, der minimerer den empiriske risiko, blot tilfældigvis passer til de set data men ikke til de hidtil usete, se figur 3.

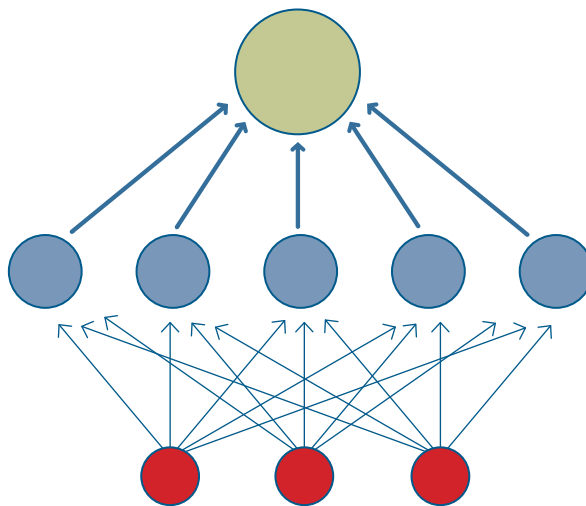
Figur 3: Træningsfejl versus generaliseringsfejl.
(a) Udviklingen af trænings- og generaliseringsfejl som funktion af størrelsen af træningssættet for en fleksibel modelklasse.
(b) Udviklingen af fejl for en mindre fleksibel modelklasse.
(c) Generaliseringsfejlen for de to modelklasser afbildet mod hinanden.



En måde at karakterisere kapaciteten af en modelklasse på er ved VC-dimensionen (Vapnik-Chervonenkis-dimensionen). Intuitivt angives antallet af frihedsgrader i modelklassen. Mere formelt udregnes dette som antallet af datapunkter, som altid vil kunne adskilles i to klasser uanset deres position. Kunsten i machine learning er at vælge en modelkapacitet, der gør, at den forventede generalisering bliver så god som muligt.

Der findes et utal af algoritmer til at indkode vores forhåndsviden, således at generaliseringen er så god som muligt, også ved en begrænset størrelse af træningssættet.

En klassisk løsning til machine learning-problemer er *neurale netværk*. Inspireret af Minsky og Paperts perceptron laves et netværk af celler, der hver summerer deres input og lader summen blive slået op i en ikke-lineær funktion, og resultatet er så input til endnu en neuron. Typisk er neurale netværk organiseret i lag med beregningerne gående fremad lag for lag (se figur 4). De blev meget populære, da back-propagation-algoritmen viste sig forholdsvis effektivt at kunne indstille vægtene, så netværket samlet minimerede træningsfejlen. Mange resultater fra og generelle metoder til machine learning er blevet udviklet i forbindelse med neurale netværk, også metoder til at indstille kapaciteten til bedst mulig generaliseringsevne. Et eksempel med et morsomt navn er "optimal brain damage", der forsøger at estimere, hvilke neuroner der skal slettes for at gøre generaliseringen bedre.



Figur 4: Neuralt netværk med et skjult lag. Data fødes ind i de røde input-neuroner. Herfra sendes de videre til de blå beregningsneuroner i det skjulte lag. Resultater herfra sendes videre til output-neuronen, der har samme funktionalitet som de skjulte neuroner: n vægtet summering af input og opslag i en ikke-lineær funktion.

Siden de neurale netværk er mange andre algoritmer blevet opfundet. Her kan blandt andet nævnes *Support Vector Machines (SVM)*. De forsøger at udvælge de datapunkter, der skal støtte løsningen, således at marginen mellem de to forskellige labels bliver så stor som muligt. Dette er et eksempel på en algoritme, der maksimerer marginen. Vapnik har vist, at optimering af marginen medfører generalisering i matematisk betydning. SVM'er har dog også vist deres store potentiale til at løse praktiske problemer og er en af de mest populære algoritmer.

En anden type algoritme er de såkaldte boosting-algoritmer. Tricket her er at have en hel lille hær af meget simple klassifikationer. Herefter anvendes de enten som flertalsafstemninger eller i en anden struktur. Meget populær er *AdaBoost*, der automatisk arrangerer de simple klassifikationer i en træstruktur, så træningsfejlen bliver så lille som muligt. Herefter kan principper til at forsøge at minimere generaliseringsfejlen også anvendes.

Et generelt trick til at estimere generaliseringsevnen er at dele data op i et træningssæt og et testsæt. Træningssættet anvendes til at vælge den bedste hypotese, og testsættet benyttes herefter til at estimere generaliseringsevnen. Dette trick kan gøres mange gange med tilfældige opdelinger i test- og træningssæt. Denne procedure kaldes *bootstrapping* og har udviklet sig som en hel disciplin for sig selv i forbindelse med udviklingen af statistiske test.

Det mest populære eksempel på bootstrapping i machine learning er nok *random forest*-algoritmen. På samme måde som AdaBoost opbygger et beslutningstræ, kan man opdele data i trænings- og testsæt. Herved kan man teste på generaliseringen af træet under dets opbygning og stoppe opbygningen, når træets generaliseringsevne ikke længere forbedres. Nu kan en ny opdeling i trænings- og testsæt tilfældigt trækkes, og derved kan et nyt træ, som er optimalt for denne splitning af data, opbygges. Processen fortsættes, indtil man har træer nok til, at man synes, man har en hel skov. Klassifikationen består så i, at træerne stemmer, og flertallet bestemmer. Fordele ved denne algoritme er, at man automatisk styrer kapaciteten af modellen. Random forest er uhyre populære i bioinformatik, hvor det efterhånden er en standardmetode til at analysere gener for association til for eksempel sygdomme.

Machine learning bruges bl.a. til at diagnosticere sygdomme

Et af de mere omtalte projekter i den seneste tid er *Learning Imaging Biomarkers* – et projekt, der meget kort fortalt går ud på at anvende billedanalyse og machine learning til tidlig diagnosticering af en lang række sygdomme, herunder bl.a. brystkræft.

Metoden går ud på at kigge efter markører eller indikatorer, altså særlige mønstre i brystvævet, der er typiske for personer, der senere viser sig at få brystkræft. Her har machine learning vist sig uhyre anvendelig, fordi man kan få computeren til at gennemregne et uendeligt antal eksempler af billeder og mønstre og herudfra skabe nogle generaliseringer om, hvorvidt der er forskelle i udseendet på vævstyperne hos hhv. personer, der får, og personer, der ikke får brystkræft.

Normalt er brystvævet hos kvinder sribet fra brystvorten og ind mod kroppen. Den nye forskning viser, at kvinder, hvis bryster viser tegn på, at sriberne går på tværs, har forøget risiko for at udvikle brystkræft. Det sribede mønster kan bruges som indikation, uafhængigt af andre faktorer, som kan læses ud af et mammografibillede.

Der er brugt en ny slags algoritme til billedanalysen, som kan genkende hundredvis af forskellige mønstre. Derefter er softwaren blevet ”optrænet” ved anvendelse af machine learning-tilgangen med mammografier fra 500 hollandske kvinder, hvor 250 på trods af en negativ diagnose senere udviklede brystkræft.

Algoritmerne er inspireret af processer i hjernen, nemlig dem der foregår i de første niveauer af den visuelle cortex, som er det yderste lag af hjernen.

Computeren kan se kvantitativt på talmaterialet, og det betyder, at den kan foretage en meget præcis mønstergenkendelse og dermed skelne kvantitativt mellem sunde mammografier og mammografier fra kvinder, som er i risikogruppen for at udvikle kræft. Det er samme type statistisk mønstergenkendelse, som kendes fra fx neurale netværk.

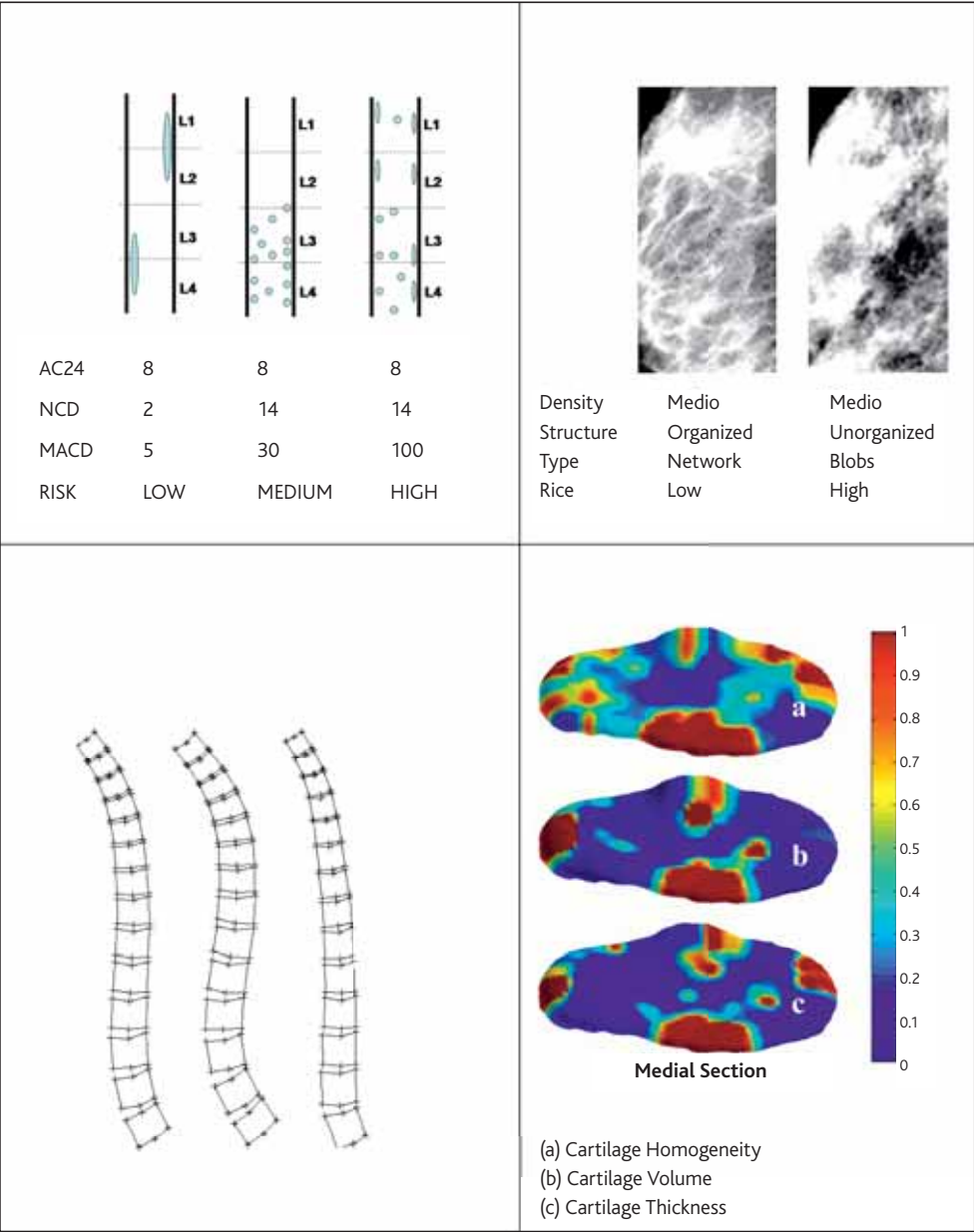
I praksis fungerer diagnosticeringen ved, at et program analyserer et mammografibillede ved at genkende, om punkter i mammografiet er en del af en tynd eller en tyk linje, en forhøjning eller en kant. Analysen korreleres med statistiske data om, hvilke kvinder der rent faktisk har fået brystkræft. Med computerens hjælp når man således frem til en hypotese om, hvilke vævsmønstre der øger risikoen for brystkræft.

Metoden anvendes også til diagnosticering af en række andre sygdomme, fx knogleskørhed, slidgigt, åreforkalkning og rygerlunger (KOL).

”Optræningen” af den anvendte software foregår på en supercomputer med meget stor regnekapacitet (læs også Brian Vinters artikel om supercomputere). Hvor det måske ville tage flere år at få en almindelig pc til at gennemløbe de millioner af billeder og afprøve de millionvis af hypoteser under optræningen, der er nødvendige for at nå frem til et pålideligt resultat, kan en beregnings-session gennemføres på få dage takket være den høje kapacitet, som forskningscomputere er udstyret med. Projektet har således brugt 1.000 af de pc-noder, som supercomputer-klyngen på H.C. Ørsted Institutet er opbygget af. Når først softwaren er ”optrænet”, kan den også bruges på en almindelig pc.

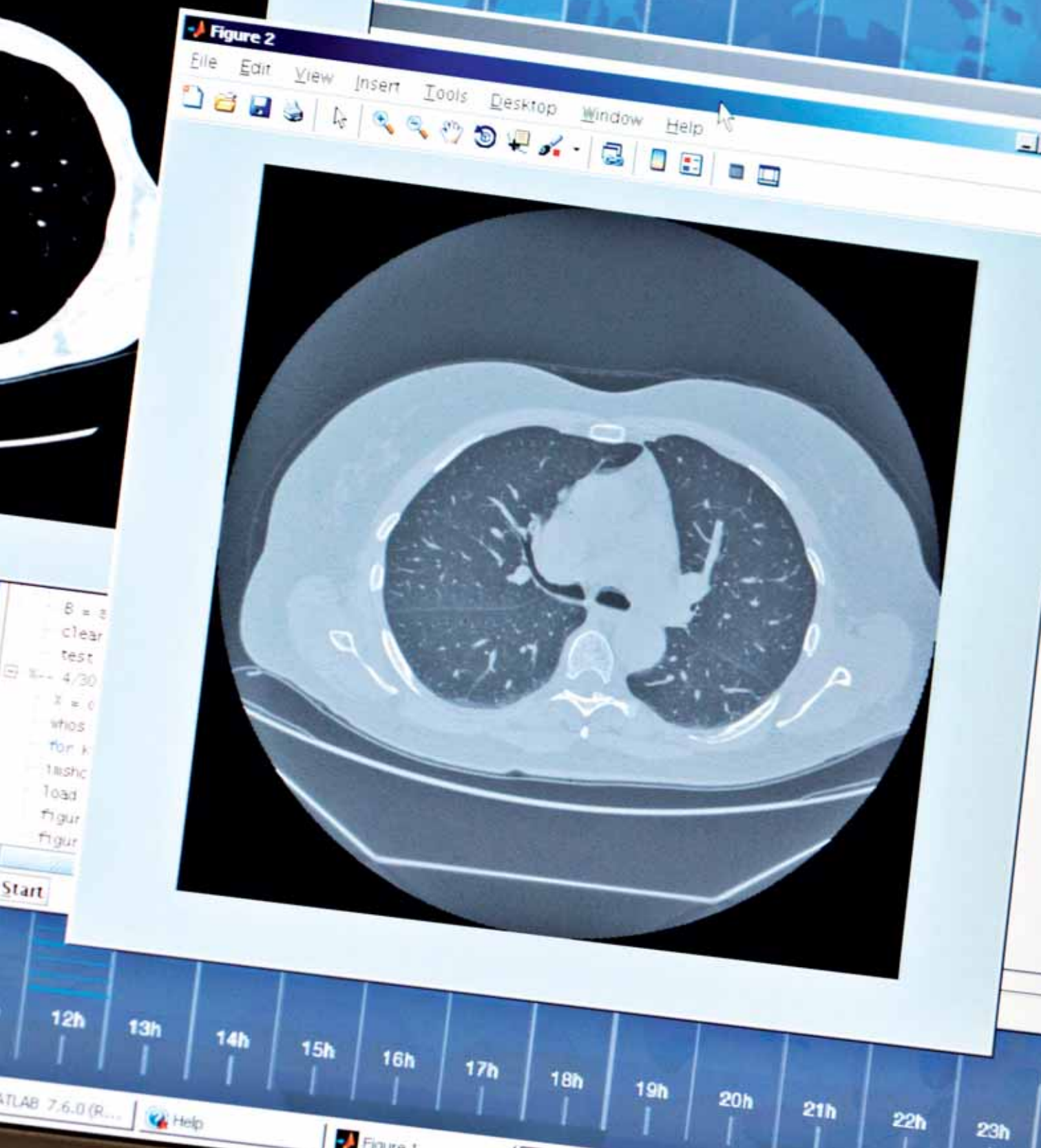
Forskningsprojektet, der i skrivende stund fortsat er i gang, er et resultat af et intensivt samarbejde mellem KU og Nordic Bioscience Imaging A/S. Det gennemføres på *eScience-centret*, der er etableret som en tværfaglig enhed med basis på H.C. Ørsted Institutet på Det Naturvidenskabelige Fakultet på KU.

Eksempler på anvendelse af machine learning til fortolkning af sundhedsdata kan ses i figur 5.



Figur 5: Første eksempel er analysen af forkalkninger i hovedpulsåren. Det viser sig, at ikke kun mængden af forkalkninger men også deres fordeling, form og størrelse spiller ind på den enkelte patients prognose. Andet eksempel viser en analyse af mammografier. Her kan man se, at den desorganiserede struktur er en indikation af brystkræft. Tredje eksempel viser genkendelsen af meget tidlige symptomer på knogleskørhed ud fra ryggens form, og endelig viser det sidste eksempel diagnose af slidgigt ud fra MR-skanninger af knæet. I alle tilfælde er machine learning anvendt til den visuelle analyse, der er vanskelig for mennesker at gennemføre kvantitativt.

Scanningbillede af lunger.
Lungen til højre er delvis
angrebet



Afrunding

Machine learning i kombination med supercomputerens kapacitet og moderne billedanalyse-værktøjer er en af moderne computerforsknings meget lovende metoder, der finder anvendelse på en lang række områder. At lade computeren finde den bedste hypotese på løsningen af et problem vil ofte både være tidsbesparende og fejlminimerende – men naturligvis er man nødt til at alliere sig med erfarne forskere på det pågældende område for at sikre sig, at man ikke stoler blindt på computerens intelligente forslag, men at der er mennesker bagved, der kan validere hypoteserne. Der er således ikke tegn i sol og måne på, at vi kommer til at opleve forsøg fra computerens side på at overtage magten over menneskene ligesom computeren Hal i Rumrejsen år 2001.

Ikke kun inden for medicinsk forskning, men også på mange andre områder vinder machine learning indpas som metode til at automatisere og forkorte processer og styrke pålideligheden af beregninger. Det gælder fx på slagterier, hvor kødudskærings- og bearbejdningsprocesserne kan effektiviseres væsentligt, inden for klima- eller vejrmødeller eller inden for maskinoversættelse og tekstbehandling for blot at nævne få eksempler. ♦

Læs mere

Hertz, J., Krogh, A. & Palmer, R.G. (1991). *Introduction to the Theory of Neural Computation*. Westview Press.

Shannon, C.E. (1948). A Mathematical Theory of Communication. *The Bell System Technical Journal*, 27, s. 379-423, 623-656, juli, oktober.

Cortes, C. & Vapnik, V. (1995). Support Vector Networks. *Journal of Machine Learning*, 20(3), s. 273-297.

MADS NIELSEN



Mads Nielsen er kandidat i datalogi fra 1992 og ph.d. fra 1995, i begge tilfælde fra DIKU. Som led i sit ph.d.-studium var han et år tilknyttet INRIA-universitetet i Sophia Antipolis i Frankrig. Herudover har han været ansat som postdoc ved Image Sciences Institute, Utrecht Universitet, Holland.

I 1996 blev han tilknyttet DIKU som postdoc parallelt med sin ansættelse i 3D-Lab under den daværende Tandlægehøjskolen (i dag SUND v. Københavns Universitet), hvor han var adjunkt i 1997-99. I april 1999 blev han den første lektor ved det nyetablerede IT-Universitetet i

København, og i juni 2002 blev han professor og leder af Image Analysis Group samme sted. I 2007 skiftede Mads tilbage til DIKU, hvor han i dag er professor og leder af Billedgruppen.

Fra en lang liste af publikationer kan fremhæves "Regularization, Scale-Space, and Edge Detection Filters" (1997), "Classification in Medical Image Analysis Using Adaptive Metric KNN" (2010) og "Distribution, Size, Shape, Growth Potential and Extent of Abdominal Aortic Calcified Deposits Predict Mortality in Postmenopausal Women" som repræsentative eksempler på forskning i anvendt matematik, datalogi og medicin.

Mads Nielsen leder en række aktuelle forskningsprojekter på disse felter.

Hjernen, den emotionelle computer?

.....
Af **Morten L. Kringelbach**, Hedonia: Trygfonden Research Group,
Oxford og Århus Universiteter
.....

Hjernen er den mest avancerede maskine, vi kender, og gennem tiderne har forskere altid benyttet den nyeste, mest avancerede teknologi til at beskrive hjernen, hvilket derfor blandt andet er overgået maskiner som eksempelvis vandpumper, vævemaskiner, telefoncentraler og computere. Det kommer dog nemt til at virke lidt absurd at sammenligne menneskehedens ypperligste, men stadig meget enkle maskiner med noget så kompliceret som hjernen.

Ikke desto mindre bliver computere stadig mere komplekse, og det virker ikke helt så absurd mere at forestille sig, at det en dag skulle blive muligt at genskabe hjernen *in silico* – og dermed genskabe os selv i vores billede. Hvis det er tilfældet, vil vi så blive i stand til at downloade vores selv og bevidsthed til en computer og leve videre som udødelige robotter?

I denne artikel vil jeg ikke forsøge at besvare så svært et spørgsmål, der stadig er mere science fiction end videnskab. I stedet vil jeg forsøge at vise, hvordan hjerneforskning er kommet tættere på at forstå mange af hjernens grundlæggende principper, og i særdeleshed hvordan hjernens afvejning af belønning og straf – og de tilhørende emotioner – er afgørende for artens overlevelse. Disse principper peger på, hvordan vi med fordel kunne forsøge at implementere tilsvarende algoritmer i computere for på denne måde at gøre computere bedre til at tage valg og tilpasse sig komplekse, foranderlige situationer.

Robothjerner

Da den amerikanske forfatter Isaac Asimov skrev sine første robotnoveller i slutningen af 1930'erne, var computere stadig analoge maskiner, som knap kunne multiplicere to tal med hinanden og konsistent få samme resultat. Anden verdenskrig og opfindelsen af transistoren ændrede med tiden denne situation så radikalt, at vi i dag har kraftige digitale computere med en regnekraft, som de færreste dengang kunne drømme om.

Teknologiske drømme er science fiction-forfatterens metier, og Asimov var som kun få andre i stand til at kombinere drømmene med en skarp prosa, som på en gang forudså og skabte en mulig fremtid.

Robothistorierne er profetiske i visionen om en fremtid, hvor menneskeheden får selskab af robotter. Robotternes fornemste opgave er at overtage alle de kedelige og farlige maskinelle opgaver, så menneskeheden kan koncentrere sig om andre og vigtigere ting.

Dermed er robotterne en slags slaver, og for at undgå ulydighed og forsøg på at overhale menneskeheden indenom er alle robotter indkodet med en etik kaldet *robotikkens tre love*. Ifølge robotikkens første lov må en robot ikke skade et menneske eller gennem mangel på handling tillade et menneske at skade sig selv. Den anden lov er underlagt den første lov og foreskriver robotter at adlyde ordrer fra mennesker, medmindre de er i modstrid med den første lov. Tredje lov tillader en robot at beskytte sin egen eksistens – medmindre det er i konflikt med første eller anden lov.

Asimovs robotter har såkaldt *positroniske* hjerner og kroppe, der nok er metalliske, men dog ikke ulig menneskets. På mange måder er disse robotter derfor skabt i vores billede. Men der er en væsentlig forskel, som går igen og igen i historierne, og det er det faktum, at robotterne alene lader sig styre af logiske regler. Dette naturligvis i modsætning til mennesker, som skildres som irrationelle væsener, som kun sjældent handler logisk.

”

*Robotter
lader sig
alene styre
af logiske
regler*

Adfærd uden følelser?

Robothistorierne forsøger på denne baggrund at undersøge de vigtige og interessante spørgsmål, som robotteknologien stiller. Som baggrund til denne undersøgelse skabte Asimov et broget persongalleri med personer som for eksempel robotpsykologen Susan Calvin, der selv er næsten lige så følelseskold som en robot. Hun er derfor glimrende til at løse de problemer, der opstår med robotter.



En DIKU-robot i aktion. Den programmeres til selv at kunne se.



Hvem siger, at det er muligt at skabe kompliceret menneskelignende adfærd uden følelser?

En klassisk historie er historien *Little lost robot*, som beskriver, hvordan en række robotter i dybeste hemmelighed bliver skabt med en modifikation af første lov. Modifikationen er blevet nødvendig, da det i forbindelse med arbejde i det ydre rum er nødvendigt for nogle astronauter at udsætte sig for en kalkuleret risiko. Dette vil ingen robot med den normale første lov tillade, og derfor har man forsøgsvis lavet en ændring.

De modificerede robotter er frygtelig arrogante, og en af dem bliver af en ilter astronaut beordret til at forsvinde. Det tager robotten alvorligt og skjuler sig blandt en række normale robotter, der ligner den til forveksling. Det er nu Susan Calvins svære job at finde robotten, inden den når at yde ubodelig skade.

Historien er en klassiker inden for science fiction-genren, da den formår at kombinere et teknologisk problem med en fin personschildring. Men samtidig peger historien også på en potentiel svaghed i Asimovs robotkoncept. For hvem siger, at det er muligt at skabe kompliceret menneskelignende adfærd uden følelser? Kunstig intelligens har gennem de sidste 50 år fejlet spektakulært med at skabe adfærd baseret på logik alene.

Biologiske hjerner er naturligvis i større (eller ofte mindre) grad i stand til at løse logiske problemer, men først og fremmest er de skabt til at overleve og reproducere sig. I en lang række videnskabelige artikler og bøger har jeg derfor argumenteret for, at hvis vi vil forsøge at skabe bevidste robotter i vores billede, har vi brug for at bygge dem med emotioner, som hjælper os med at afveje belønning og straf, hvilket i sidste ende tillader os at overleve og reproducere os.

Læringsnetværk

Hjernen består grundlæggende af neurale netværk, som i sidste ende er grundlaget for vores mentale evner og intelligens. Det fundamentale princip for læring i neurale netværk blev foreslået af den canadiske psykolog Donald Hebb i 1948 og er ganske enkelt, at synapser kan ændre deres styrke, således at de påvirker andre neuroner med enten mere eller mindre styrke (se faktaboks om neuroner).

Hjernen ville ikke være i stand til at fungere uden gigantiske neurale netværk, og det er derfor vigtigt at forstå lidt af disse netværks funktion. Vi begyndte først at forstå, hvor kraftfulde disse neurale netværk kan være, da vi fik mulighed for at simulere dem på computere, hvilket begyndte med den elektroniske computers fødsel i 1940'erne og for alvor tog fart med de sidste årtiers eksplosion af computerkraft.

Der er grundlæggende forskel på hjerner og computere. Hjerner er blevet flikket sammen nærmest hulter til bulter af diverse komponenter fra den udviklingshistorie, som vi deler med andre dyr. I modsætning hertil er vore dages computere blevet designet og konstrueret efter strenge logiske principper, der svarer til en såkaldt universel turingmaskine, som først blev beskrevet af den engelske matematiker Alan Turing i starten af 1930'erne (se artiklen af Torben Mogensen). Dette videnskabelige arbejde blev siden videreført i 1940'erne af den amerikanske matematiker John von Neumann, der fandt på, at man kunne gemme programmer på logiske maskiner, der siden blev til vores digitale computere. Den første digitale computer blev i øvrigt skabt af John

Om neuroner

Neuroner er nogle af hjernens mindste, men samtidig funktionelt vigtigste dele. En neuron fungerer basalt som en lille analog computer, der bliver aktiv, hvis summen af den modtagne elektriske aktivitet er højere end en tærskelværdi. Neuroner forbinder sig med andre neuroner og danner neurale netværk, der er basis for al mental aktivitet.

Men hvordan fungerer denne mindste enhed? Man kan bruge dominobrikker som en simpel metafor for neuroners aktivitet, da en samling af neuroner kan skabe en dominoeffekt, hvormed en neuron kan starte en kaskade af aktivitet. Aktiviteten bevæger sig fra én neuron til den næste i rækken, ganske som dominobrikker falder, den ene efter den anden. Metaforen bryder imidlertid let sammen, da neuroner er ganske særlige dominobrikker, der i modsætning til almindelige af slagsen selv skal kunne stille sig selv op igen – og igen. Ikke nok med det, men neuroner kan også i modsætning til dominobrikker selv afgøre, både om og i givet fald hvornår de skal få de næste brikker til at falde.

Den vigtigste centrale egenskab ved neuroner er, at de er i stand til at lære. En neuron kan selektivt ændre sin indflydelse på de næste neuroner, hvilket svarer til, at neuroner kan bestemme, hvilke af de andre dominobrikker der skal påvirkes mest.

En mere detaljeret beskrivelse af de enkelte neuroners aktivitet og struktur skelner mellem fire væsentlige delkomponenter: *dendrit*, *soma*, *akson* og *synapse*, se figur 1. Synapsen er bindeledet mellem neuroners *aksoner* og *dendritter* og stedet, hvor elektriske signaler bliver konverteret til kemiske signaler over den såkaldte synaptiske kløft. Dernæst bliver signalet straks konverteret tilbage til et elektrisk signal på den næste neurons dendrit.

Output fra en neuron kommer via akso-

ner og bliver til input til andre neuroner via synapser, der sidder på disse neuroners dendrit eller rettere *dendritiske træ*, da neuroner har en vildtvoksende træliggende struktur, som fungerer som et stort flodsystem til neuronens indre, soma. Det dendritiske træ er således inputdelen af neuronen og sikrer, at neuronen får kontakt med tilstrækkelig mange andre neuroner til, at den kan foretage sig noget fornuftigt. Metaforisk kan man sige, at i *soma* samles de elektriske strømme, og hvis summen heraf er over en given tærskelværdi, så kontaktes andre neuroner, ved at neuronen via sine aksoner og synapser affyrer et elektrisk signal kaldet aktionspotential.

Der findes mange forskellige typer neuroner med mange forskellige former og egenskaber i det centrale nervesystem i hjernen og i det perifere nervesystem uden for hjernen, se figur 2. For eksempel er neuroner, der modtager sanseindtryk fra huden eller tungen, meget forskellige fra de neuroner, der styrer vores motoriske bevægelser fra det motoriske område og rygmærven. Grundlæggende kan neuroner inddeles i to typer: *aktiverende* og *hæmmende*. De *aktiverende neuroner* er langt i overtal og udgør ni tiendedele af alle neuroner i hjernebarken. De modtager synaptisk input via tornelignende gevækster på deres store dendritiske træer, har forbindelser med både lokale og fjerntliggende neuroner og har derfor ofte aksoner i den hvide hjernemasse.

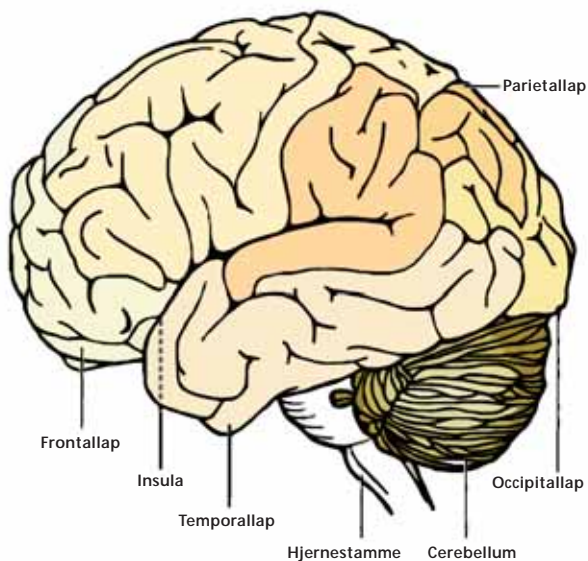
De *hæmmende neuroner* har derimod ikke tornelignende gevækster, men modtager synapser direkte på deres dendritiske træ, har kun lokale forbindelser og har som hovedregel ikke aksoner i den hvide hjernemasse. Denne type neuroner påvirker andre neuroner negativt og har dermed en hæmmende virkning på neural aktivitet.

Hvis der kun fandtes neuroner med positiv eller aktiverende effekt i hjernen, ville der hurtigt opstå fuldstændig

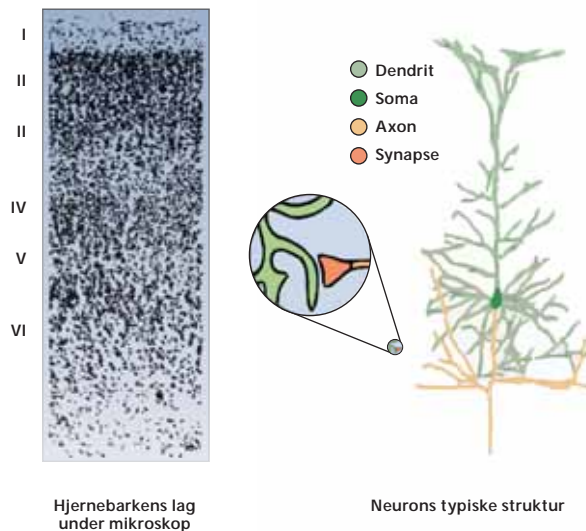
ukontrollabel aktivitet. Det sker alligevel undertiden i epileptiske anfald, men er heldigvis sjældent. Det er derfor vigtigt, at neuronerne ikke bliver overstimuleret, og at aktiviteten af neurale netværk forbliver under kontrol via indflydelsen fra de hæmmende neuroner.

Synapser og dermed den synaptiske kløft fungerer som skitseret ovenfor som det sted, hvor information fra neuroner for en kort stund bliver til kemiske i stedet for elektriske signaler, og hvor læring kan finde sted. Det kræver energi og et meget stort cellulært maskineri at understøtte disse processer til at overføre information. Synapseoverførsel er derfor ikke blot en mindre detalje, men viser sig at have stor betydning, da det også giver mulighed for, at signalstoffer kan påvirke virkemåden for mange neuroner på én gang ad biokemisk vej. Der findes som bekendt en lang række væsentlige signalstoffer, der påvirker hjernen på forskellig vis, og som vi har set spiller en stor rolle blandt andet i en lang række psykiske problemer som depression, skizofreni, psykoser og angstneuroser.

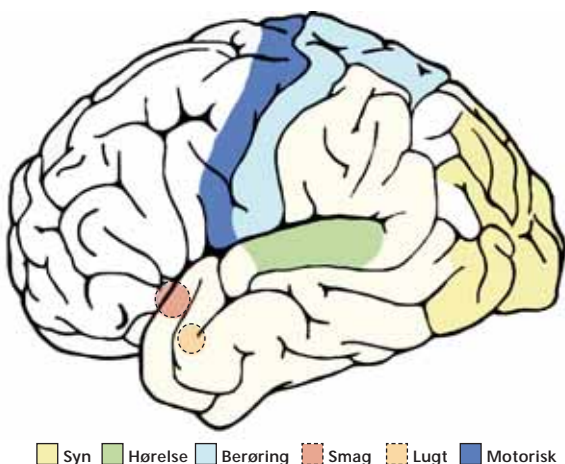
Neuroner er dermed ganske avancerede små maskiner, der kan summere aktivitet fra andre neuroner og afgøre, om denne aktivitet skal videreføres til endnu andre neuroner. Dette billede af neuroners funktion er naturligvis forsimplet og udelader i sagens natur en mængde detaljer. Til trods for at Alan Hodgkin og Andrew Huxley allerede i 1963 modtog Nobelprisen for deres grundlæggende matematiske beskrivelse af neuroners dynamiske struktur, så opdager vi stadig nye, interessante detaljer om neuroners funktion. Det er for eksempel for nylig blevet opdaget, at aktionspotentialet tilsyneladende ikke kun flyder ud via aksonerne, men også flyder tilbage op ad det dendritiske træ og på den måde påvirker neuronens læring, men disse spidsfindigheder ændrer ikke grundlæggende på den ovennævnte beskrivelse.



Figur 1: Hjernens geografi. Hjernen består først og fremmest af hjernebark, cerebellum og hjernestamme. Hjernebarken kan opdeles i større og mindre områder, som vist på figurens venstre del. Denne opdeling i områder kan ske på baggrund af en undersøgelse af hjernebarken under mikroskop, hvilket vi-



ser hjernebarkens lag, der er forskellige i hvert mindre område (i midten). Hver af de sorte pletter, man kan se i hjernebarken under mikroskop, svarer til en endnu mindre del, en nerve-celle eller neuron, der er grundlaget for al hjerneaktivitet og dermed alle vores nydelser.



Figur 2: Hjernens områder for sansning og bevægelse. Der er specialiserede primære områder i hjernebarken, hvor sansninger bliver afkodet. På figuren er de fem vigtigste sansninger markeret, syn, hørelse, berøring, smag og lugt, men det er naturligvis ikke de eneste sanser, da der også findes andre sansereceptorer, der kan give hjernen vigtig information om, for eksempel hvor opspilet maven er (hvilket spiller en åbenlys rolle i forbindelse med sult). Men som altid foregår det virkelig interessante i de store, hvide pletter på hjernens landkort – i de områder, man kalder højere-ordens-associations-områder. Her foregår den aktivitet, der hjælper med at træffe emotionelle valg, der afgør, om kroppen skal i bevægelse, eller der skal ske ændringer i kroppens indre miljø.

Vincent Atanasoff i slutningen af 1930'erne, men som så mange andre pionerer blev Atanasoff stort set glemt af eftertiden.

Disse forskelle i design og historie betyder, at hjerner er gode til nogle ting og mindre gode til andre, på samme måde som computere. Menneskehjerner er (forholdsvist) gode til at få mennesker til at overleve, reproducere sig og træffe valg i meget komplekse omgivelser, men er mindre gode til eksakte matematiske beregninger, da det sjældent er nødvendigt for at overleve.

Computere har ikke behov for at overleve, har så vidt vides endnu ikke reproduceret sig og bliver sjældent bedt om at vælge. Til gengæld er computere fantastiske til at lave endeløse præcise beregninger. Disse beregninger er imidlertid ikke nødvendigvis særlig sofistikerede, og man kan i særdeleshed vise, at den universelle turingmaskine, der er grundstenen for computere, kun kan løse bestemte klasser af problemer. Visse problemer kræver så mange beregninger, at de ikke kan løses fuldstændigt af selv den største computer inden for en overskuelig fremtid. Et eksempel på et sådant svært problem er at finde den billigste rejserute mellem forskellige byer for en rejsende forretningsmand, når rejserne mellem byerne koster forskellige pengesummer alt efter afstanden. Når der er mere end blot en håndfuld byer, vokser beregningstiden hurtigt så voldsomt, at selv kraftige computere ikke kan løse problemet inden for en menneskealder.

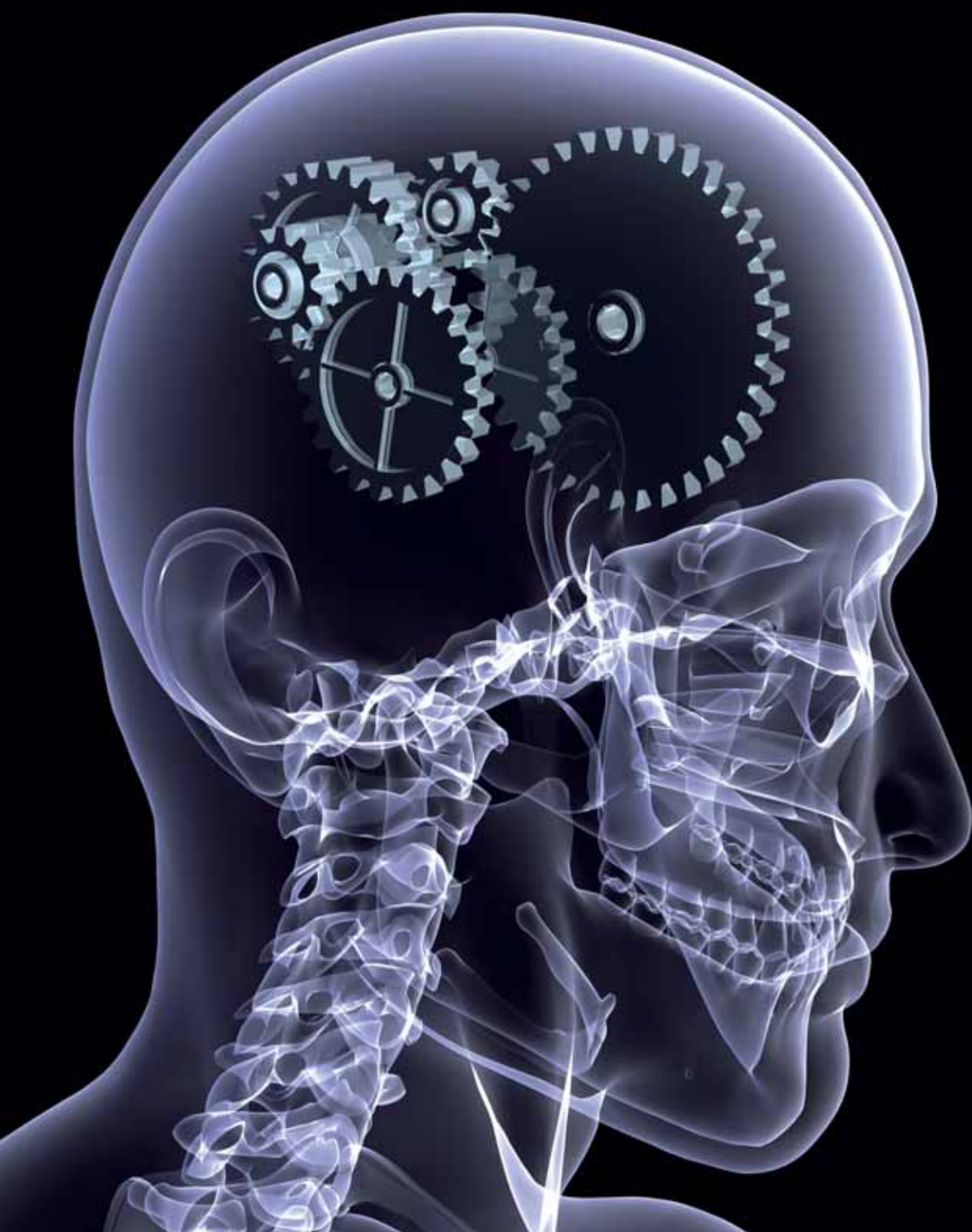
Men rejsende forretningsmænd tager alligevel fra by til by, og snart sagt alle de problemer, som hjernen løser hele tiden (som for eksempel det at *se*), er så svære beregningsmæssigt, at computere ikke kan være sikre på at have fundet den bedste løsning selv med uendelig tid til rådighed. Hjernen løser dog heller ikke disse problemer fuldstændigt, men bruger i stedet en lang række tricks og tommelfingerregler, der i stedet kommer med en passende tilnærmelse på en løsning, som derfor ikke altid er det bedste valg.

Bemærk, at hvis man kunne afdække alle disse tricks (og det er et stort *hvis*), så er der principielt intet til hinder for, at en computer også kan lære dem og dermed løse problemet lige så godt. Der er ikke nogen god beregningsmæssig grund til, at man ikke skulle kunne efterligne eller simulere visse dele af hjernens funktion på computer og dermed løse de samme problemer mindst lige så godt.

En fuldstændig beskrivelse af, hvad vi ved hjælp af computere har lært om egenskaberne i de neurale netværk i hjernen, er beskrevet i læseværdige detaljer andetsteds. Her vil jeg kort beskrive fremskridtene i at forstå grundlaget for de valg, som vores hjerner må tage hele tiden, for på den måde at undersøge, om det kan være muligt at få computere til at løse de samme problemer og dermed måske tilmed blive bevidste. Det er derfor vigtigt at vide, hvad studiet af neurale netværk har bidraget med til vores forståelse.

Neurale netværk i silcium

De første mange år var det minimalt, hvad forskerne lærte om hjernens neurale netværk. De kaldte godt nok deres netværk for neurale, men de havde meget lidt tilfælles med rigtige neuroner. Et af de første forsøg med kunstige neurale netværk blev lavet i computerens spæde barndom i 1943 af Warren McCulloch og Walter Pitts på computere, der var på størrelse med huse og havde



regnekraft langt under en af vore dages regnemaskiner. Det var derfor i sig selv en bedrift, at det lykkedes forfatterne at skabe en neural struktur med neuroner, som kunne fungere som logiske netværk, der kunne beregne simple logiske operationer som OR og AND. De neurale netværk viste sig desværre, på linje med den tids analoge computere, ikke at være særlig tolerante for fejl, og der gik derfor en rum tid, før de neurale netværk blev undersøgt i større detaljer.

Den næste type neurale netværk var den såkaldte 'perceptron', der blev bygget i 1950'erne som lag af neuroner forbundet med synapser, og som både blev simuleret i computere og bygget som fysiske enheder af i hundredvis af radiorør. Perceptronen var ikke alene i stand til at sammenholde allerede kendte input med et output, men kunne også generalisere fra støjfyldte input til meningsfyldt output. Det betød, at perceptronen ligesom hjernens neurale netværk er fejltolerant, og mange forskere anså perceptronen for et stort skridt fremad.

Man blev dog hurtigt klar over, at fremskridtet var til at overse, da det i slutningen af 1960'erne blev vist, at perceptronen ikke var i stand til at implementere en meget simpel logisk operation kendt som enten-eller ('XOR'). Denne logiske operation betyder, at hvis en maskine har to binære input, der enten kan være sande eller falske, så skal maskinen lære kun at svare sandt, hvis de to input er forskellige. Det vil sige, at svaret kun er sandt, når det ene af de to input er sandt, men falskt, når de begge to enten er sande eller falske.

Selv for entusiaster var det svært at se, hvordan hjernen skulle kunne fungere uden at være i stand til at løse dette simple problem, og interessen for neurale netværk kølnedes betydeligt i en årrække.

Kunstige neurale netværk

Det var først i midten af 1980'erne, at kunstige neurale netværk fik en renæssance. To vigtige opdagelser fik dampen tilbage på kedlen. Dels viste det sig, at hvis man har et netværk med tre forbundne lag af neuroner, hvor det andet lag er skjult fra input, så kan man ved hjælp af en snild såkaldt 'back-propagation'-algoritme lære det neurale netværk at udføre den XOR-funktion, som perceptronen ikke kunne. Som så ofte før i videnskab var denne opdagelse ikke fuldstændig ny, men var blevet demonstreret tidligere i mere obskure videnskabelige tidsskrifter. Det har siden vist sig, at det er højst usandsynligt, at hjernen benytter denne algoritme, men opdagelsen skabte alligevel en begejstring for neurale netværk blandt kognitive psykologer, som ikke siden er aftaget synderligt.

Den anden vigtige opdagelse blev gjort af den amerikanske fysiker John Hopfield, der fandt en elegant matematisk formulering for såkaldt 'selv-associative' neurale netværk. Disse selv-associative neurale netværk, hvor neuroner har massive forbindelser til både alle andre neuroner og til sig selv, viste sig at være en god model for, hvordan hukommelsesdata kan gemmes i hjernen. I særdeleshed benytter modellen såkaldt hebbiansk læring til at gemme mange hukommelsesspor oven på hverandre. Ganske som man ser det i hjernen, så er disse hukommelsesspor forbløffende resistente over for tab af selv mange af netværkets enkelte neuroner.

Samtidig gav Hopfields matematiske beskrivelse fysikere, dataloger og matematikere en mulighed for at lade sig konvertere til hjerneforskere, hvilket var tiltrængt, for det var nedlæggelsestider for de store kernefysiske initiativer, der ellers havde holdt mange af dem beskæftiget. Måske

blandt andet derfor er Hopfields oprindelige artikel fra 1982 blandt de mest citerede artikler i hjernevidenskab. Det, til trods for at det fra starten var tydeligt, at der også var en lang række mindre plausible biologiske egenskaber ved modellen. For eksempel kan Hopfields neuroner både hæmme og aktivere andre neuroner, hvilket som nævnt ikke er tilfældet for neuroner i hjernen, der enten er hæmmende eller aktiverende.

Disse to vigtige opdagelser inden for neurale netværk har derfor (som meget ny forskning) en række åbenlyse begrænsninger, men gav alligevel anledning til mere interessant videnskabelig forskning i særdeleshed i brugen af computersimuleringer til at bekræfte og forudsige egenskaber ved hjernens neurale netværk. Grundlæggende er det lønlige håb for mange forskere, at denne abstraktion af hjernens neurale netværk en dag vil give mulighed for at skabe en hjerne i en computer, som man så vil kunne tænde og slukke for efter behag. Denne forskning har dog stadig langt igen, og selv adfærden hos meget simple organismer som rundorme med kun 302 neuroner er stadig ikke blevet succesfuldt simuleret i alle aspekter.

Kunsten ved disse computersimuleringer er at bestemme, hvor meget af hjernens overvældende kompleksitet der skal medtages for, at resultaterne kan sige noget interessant om hjernens funktion. Nogle benytter supercomputere til at genskabe kompleksiteten af en enkelt neuron helt ned til det molekylære niveau, mens andre genskaber gigantiske netværk med simple binære neuroner til at forsøge at efterligne hjernens funktion.

Vi er nu bedre i stand til at vurdere, hvordan hjernens rigtige neurale netværk træffer beslutninger. Hjernescanninger viser den makroskopiske aktivitet af disse neurale netværk i menneskehjernen. Det er dog vigtigt at huske, at hjernescanningsteknikker som funktionel magnetisk resonans (fMRI) og positron-emissions-tomografi (PET) kun er meget overordnede mål, der viser *gennemsnitlige* værdier af den neurale aktivitet i store hjerneområder målt ved den mængde blod (og dermed ilt og sukker), der bliver tilført hjerneområder over sekunder. Disse mål kan komplementeres med andre metoder som magnetoencefalografi (MEG), der måler den neurale aktivitet direkte over millisekunder, men som stadig kun er et gennemsnit af aktiviteten af titusinder af neuroner. Alternativt kan man benytte elektroder til at lytte med på enkelte neuroners aktivitet, hvilket til gengæld ikke fortæller meget om hele netværkets aktivitet og sjældent kan lade sig gøre på mennesker.

Rationelle handlinger?

Alle former for valg må kræve afvejninger af de forskellige muligheder for belønning og straf, hvilket i bedste fald fører til gode rationelle valg. Mange mennesker vil mene, at deres bevidsthed er kraftigt medvirkende til at tage sådanne frie rationelle valg, der styrer, om ikke hele deres liv, så i hvert fald det meste. Men det er også en mulighed, at denne intuition er en illusion, som meget af vores såkaldte selvindsigt.

Lad os starte med rationalitetsbegrebet, der naturligvis er et næsten uopnåeligt ideal, da det i sin definition kræver, at man indhenter al den viden, der skal til for at tage det bedste valg. Man vil aldrig have tid nok til at indhente den nødvendige viden, og selv med uanede mængder tid er det

umuligt at afgøre, hvornår den nødvendige information er til stede. Selv hvis man slækker på kravene om perfekt information, så har psykologiske forsøg vist, at mennesker sjældent tager megen notits af den information, der trods alt er til rådighed. Evolutionen har konstrueret vores hjerner således, at selv når vi ikke er under indflydelse af rusmidler, træthed eller stærke følelser, så er vores valg ofte dybt irrationelle.

Langt de fleste ville for eksempel vælge at gå ganske langt for at spare 5 kroner på en genstand, der koster 10 kroner rundt om hjørnet. Men snart sagt ingen ville gå særlig langt for at betale 9.995 kroner for en genstand, der koster 10.000 kroner om hjørnet. Det ville ikke desto mindre være det mest rationelle valg, da det jo drejer sig om de samme 5 kroner.

Der findes en lang række andre eksempler på rationalitetens sammenbrud hos almindelige mennesker, og det vil måske være mere korrekt at tale om, hvordan mennesker opfører sig *fornuftigt* det meste af tiden. Men er det en følge af *bevidst* fornuft?

Emotioner og bevidst fri vilje?

Nogle hjerneskadete patienter er ude af stand til at handle, som de fortæller, at de godt ved, at de burde. Men meget tyder på, at vi måske nok kan begrunde de fleste af vores handlinger, men at disse begrundelser ikke har meget at gøre med, hvad der egentlig skete, da vi skulle vælge adfærd, og derfor i bund og grund er efterrationaliseringer, der foregår efter handlingen.

En fortolkning af de videnskabelige data er, at handlinger udspringer af ikke-bevidste processer, som bevidstheden først siden fortolker. De østrigske neurofysiologiske forskere Kornhuber og Deecke viste, at op til 2 sekunder før vi bevæger en finger, er der allerede neural aktivitet, *beredskabspotentiale*, i den forreste del af vores hjerner.

Den amerikanske neurofysiolog Benjamin Libet kendte til disse forsøg, men forsøgte at bygge videre på dem ved at bede forsøgspersoner om at bevæge deres finger, når de har lyst. Forsøgspersonerne bliver dernæst bedt om at fortælle, hvor viseren stod på et meget hurtigt roterende ur, da de følte trang til at bevæge fingeren. Det viste sig ikke overraskende, at denne trang optræder *før* bevægelsen (cirka et halvt sekund før), men *efter* den neurale aktivitet forbundet med beredskabspotentialet. Det betyder med andre ord, at der er hjerneaktivitet relateret til fingerbevægelsen mindst et halvt sekund til et helt sekund, før vi bliver bevidste herom, og at denne hjerneaktivitet derfor i sigens natur må være ikke-bevidst.

Det betyder altså, at vores frie vilje tilsyneladende synes at udspringe af ikke-bevidste processer, hvilket har fået nogle forskere til at mene, at fri vilje ikke eksisterer. Men det følger ikke nødvendigvis, for hvis man accepterer, at vi kun har indsigt i en meget lille brøkdel af vores hjerneprocesser, så følger det, at vores valg sagtens kan være frie, uden at bevidstheden er involveret.

Men denne fortolkning kan også betyde, at *bevidst* fri vilje er en illusion, og at ubevidste hjerneprocesser styrer større dele af vores handlinger, end vi almindeligvis oplever subjektivt – og dermed spiller en stor rolle i konstruktionen af selvet.

Præferencer og intuitioner opnået på baggrund af emotionel læring er grundlæggende i denne konstruktion. Vi parrer hele tiden adfærd med belønning og straf, og brugen af ikke-bevidste hjerneprocesser til denne proces skaber muligheder for at undgå flaskehalse i beslutningsprocesser.

Bevidste hjerneprocesser er serielle og langsomme og består sjældent af mere end 5-9 elementer på et givent tidspunkt. Dette i modsætning til de ikke-bevidste hjerneprocesser, der giver mulighed for at behandle selv meget store informationsmængder samtidig. Denne massive ikke-bevidste informationsbehandling betyder, at vores valg påvirkes af information fra sanserne, der ikke normalt er tilgængelig for bevidstheden som for eksempel subliminale visuelle stimuli af meget kort varighed.

Vores fornemmelse af bevidst fri vilje er derfor paradoksalt i den forstand, at det er svært at se, hvordan den kan opstå i vores biologiske hjerner, men samtidig er det tydeligvis nødvendigt for mennesker at føle, at vi har en fri vilje og dermed kontrol over vores handlinger – også selvom vores bevidste selv ikke nødvendigvis er meget andet end blot en passager. Vores emotioner ligger som en konstant understrøm af omfattende ikke-bevidste hjerneprocesser, der er en afgørende del af vores bevidsthed, da det giver os et 'drive' til at opretholde livet, til at reproducere os og dermed til stadig bevægelse.

Emotion kommer fra latin, og betyder "det, som bevæger os til handling", og dybest set er det ansporingen til handling, der er evolutionens mening med emotioner og dermed giver en forklaring på, hvorfor det altid føles som noget at være bevidst.

Emotionelle, bevidste computere?

I denne artikel har jeg forsøgt at skitsere den begyndende forståelse af hjernens underliggende principper og algoritmer til afvejning af belønning og straf i forbindelse med valg, hvilket peger på en række nye, spændende muligheder for datalogi. Det synes åbenlyst, at mange af disse algoritmer vil kunne bruges til at gøre computere bedre til at behandle endog meget komplekse datasæt. Vi er på tærsklen til at kunne skabe emotionelle computere, der på meget mere direkte vis vil kunne tilpasse sig. Det medfører naturligvis ikke, at disse computere vil være bevidste og have bevidst fri vilje. Men det betyder, at vi snart vil være i stand til at skabe meget mere avanceret, adaptiv adfærd, hvilket uden tvivl vil få meget væsentlig indflydelse på vores omgang med computere.

Allerede nu har vores voksende forståelse af hjernens grundlæggende principper gjort det muligt at ændre i menneskehjernens funktion. Vi indopererer rutinemæssigt elektroder i menneskehjerner, der ændrer på hjernens neurale aktivitet, hvilket fundamentalt kan ændre på adfærd (se fx videoerne af patienter med bevægelsessygdomme og kronisk smerte på www.kringelbach.dk/nrn).

Hvad der dermed er mere klart end nogensinde før, er, at datalogi er helt central for denne hjerneforskning, som allerede nu hjælper mange mennesker, som vi ikke kunne hjælpe før. Med tiden får vi stadig større indsigt i hjernens forunderlige kompleksitet og dermed i menneskets grundlæggende nydelsesfulde natur. Det er en spændende rejse, som er godt undervejs, men som i særdeleshed godt kunne bruge flere dataloger til at hjælpe med at kortlægge bevidsthedens geografi. ♦

Læs mere

Kringelbach, M.L. (2009) *The Pleasure Center*. Oxford University Press.

Kringelbach, M.L. & Aziz, T.Z. (2008) Sparking recovery with brain "pacemakers".

Scientific American Mind, 6, s. 36-43.

Kringelbach, M.L., Jenkinson, N., Owen, S.L.F. & Aziz, T.Z. (2007) Translational principles of deep brain stimulation. *Nature Reviews Neuroscience*, 8, s. 623-635.

MORTEN KRINGELBACH



© Isak Hoffmeyer

Morten L. Kringelbach, professor og Dr.phil., er hjerneforsker og leder af Hedonia: Trygfonden Research Group, der er en transnational forskningsgruppe mellem Oxford og Aarhus Universiteter. På baggrund af sin pionerforskning i nydelse, bevidsthed, dyb-hjerne-stimulation og hjerneskaning er Morten

Kringelbach både "fellow" af Association for Psychological Science og med i advisory boardet for *Scientific American*. Han har en kandidatgrad i datalogi fra DIKU i 1997 og har skrevet et par hundrede videnskabelige og populærvidenskabelige artikler foruden en lang række

bøger, senest "Den nydelselsfulde hjerne" (Gyldendal, 2008), "The Pleasure Center" (OUP, 2009) og "Pleasures of the Brain" (OUP, 2010).

Morten giver ofte interview i aviser, tv og radio (fx The Guardian, Alt for Damerne, DR, TV2, BBC og Discovery Channel). Han har også givet mange videnskabelige og offentlige foredrag nationalt og internationalt (fx Statens Museum for Kunst, Louisiana, OECD, EU). Senest har han holdt foredrag om sin forskning ved Nobel Forum i Karolinska Institutet, Stockholm. Han er særlig interesseret i krydsfeltet mellem kunst og videnskab og laver skulpturer i samarbejde med kunstneren Annie Cattrell, hvoraf flere nu er på museum. Kringelbach fik tildelt Forskningskommunikationsprisen 2006 af HKH Kronprinsesse Mary.



MENNESKERS OMGANG MED DEN DIGITALE TEKNIK





Menneskers omgang med den digitale teknik

.....
Af **redaktionen**
.....

Den digitale verden er i dag en del af de fleste menneskers dagligdag. Når mennesker bruger deres mobiltelefon, ser tv, søger oplysninger på Wikipedia, underholder sig med computer-spil eller downloader musik, bruger de den digitale verden til en lang række daglige gøremål.

Den digitale verden er lige så reel som den fysiske verden, og i takt med at det bliver muligt at interagere fysisk med den digitale verden, bliver det vanskeligere og vanskeligere at skelne mellem de to. Der er imidlertid en forskel. Den digitale verden er en modelverden. En ordnet og systematisk verden, som vi kan bruge som et redskab i vores tilværelse, men også manipulere med og forvrænge, hvis dette er vores hensigt. Artiklen om informationssikkerhed af Peter Blume giver et indblik i de mange sikkerhedsmæssige faldgruber, den digitale verden også frembyder, og kommer med bud på, hvordan man bl.a. ad lovgivningens og teknologiens vej kan forebygge sikkerhedsrisici.

Vi kan udbygge og forbedre teknologien, så vi kan udnytte den digitale verden til stadig flere opgaver, men der er også forhold, som ikke kan inddrages, idet de simpelthen ikke er mulige at beskrive i form af data på en ordnet og systematisk måde. Menneskers følelser og adfærd over for medmennesker kan ikke på nogen rimelig måde udtrykkes i data – simpelthen fordi de ikke er entydige, men vekslende og modstridende. Inden for litteraturen og i musikken spiller man netop på denne mangfoldighed. Et værk skal kunne fortolkes og udlægges af forskellige mennesker. Denne side af den menneskelige tilværelse er der ikke plads til i den digitale verden.

Man kan sige, at vi med opbygningen af den digitale verden har realiseret Diderots vision om at bringe verden på orden og sætte naturen i system. Den store udfordring bliver nu at få indsigt i, hvad vi kan bruge denne verden til, og hvorledes den skal suppleres og udvikles. På trods af sine mange muligheder har den digitale verden også sine begrænsninger, og man skal være påpasselig med udelukkende at søge tekniske løsninger på problemer, som, når det kommer til stykket, bedst løses ved at påvirke og ændre menneskers adfærd. I sidste instans er det også menneskers adfærd, der vil være bestemmende for, hvorledes den digitale verden skal udvikles, og hvorledes den skal bruges.

Den menneskelige adfærd er da også udgangspunktet, når nye generationer af digitale systemer udvikles. Med mennesket som prøvekanin gennemføres der verden over laboratorieeksperimenter for at finde frem til de mest ergonomiske, de mest logiske eller de mest smarte pro-

grammer, terminaler, tastaturer, mus og skærmtyper, som samtidig skal modsvare den gængse opfattelse af, hvad der er godt design.

Også her skal der være plads til individuelle krav og ønsker – nogle sværger til Mac, mens andre er inkarnerede Windows-fans. Nogle er til genvejstaster, mens andre partout skal have en mus i hånden. Er Apples nye iPad en dødssejler eller den nye computerstandard? Det vil tiden vise – forskningen kan give ret præcise data om den historiske udvikling til dato, men kun gætte på fremtiden. Er musen død?, spørger Esben Warming Pedersen i sin artikel om tangible computing, der gennemgår de mest gængse brugergrænseflader samt giver en status på den nyeste teknik anno 2010.

I den sidste artikel i dette afsnit belyser Rasmus Helles fremkomsten af de digitale medier, som også i høj grad er et produkt af menneskets adfærd og dets valgmuligheder samt de sociale konventioner, ethvert menneske i en given kultur er underlagt. ❖

Tangible computing: musens død?

.....
Af **Esben Warming Pedersen**, DIKU
.....

Da Doug Engelbart i 1968 præsenterede verdens første mus (figur 1) under en fremvisning på Stanford Research Institute, var der næppe nogen af tilhørerne, der havde fantasi til at forestille sig, hvor stor betydning dette lille værktøj ville få. I dag, 42 år efter, er musen et af de mest benyttede værktøjer verden over, og de fleste mennesker betjener musen med så stor rutine, at de ikke længere tænker over, hvilke bevægelser de gør med hånden.

Menneskets evne til at udvikle og benytte værktøjer er en af de egenskaber, der adskiller os fra jordens øvrige arter. Gennem tiden har mennesket udviklet et hav af forskellige værktøjer, som alle hjælper os med at løse forskellige opgaver. Stenaldermanden havde sin stenøkse, når han skulle hugge, vikingen have sit drikkehorn, når han skulle drikke, og den opdagelsesrejsende havde sin sekstant, når han skulle navigere. Fælles for de nævnte værktøjer er, at de alle hver især tjener ét enkelt og specifikt formål, og stenaldermanden ville næppe finde den opdagelsesrejsendes sekstant særlig bevendt.

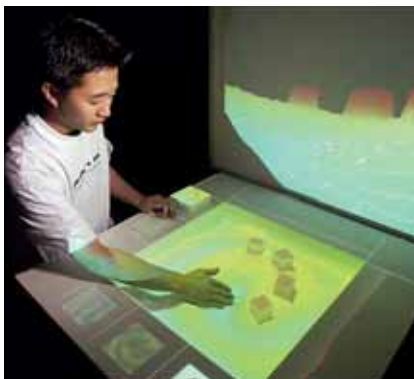
Musen bliver derimod, trods sin forholdsvis begrænsede funktionalitet, anvendt til meget forskelligartede opgaver inden for vidt forskellige domæner. Den samme mus kan altså sagtens bruges til at surfe, spille computerspil, male, skabe 3d-animationer og komponere elektronisk musik med. Men er musen nødvendigvis det bedste værktøj til alle disse opgaver? Er det overhovedet muligt at gengive detaljen og lethed i en malers penselstrøg med en mus? Kan en elektronisk musiker udtrykke sig tilstrækkelig udtryksfuldt under en koncert ved at pege og klikke? Det er sådanne spørgsmål, forskningen inden for tangible computing prøver at besvare. Ved at designe nye typer af brugergrænseflader og benytte avancerede interaktionsteknikker skaber forskere computersystemer, der er så naturlige at operere, at brugeren glemmer, at det er en computer, der benyttes.

Tangible betyder direkte oversat taktil – dvs. "håndgribelig" eller "som har at gøre med berøringssansen". I tangible user interfaces benyttes fysiske håndgribelige objekter i interaktionen, og brugeren kan derved trække på erfaringer fra den fysiske verden. Vi har som mennesker en utrolig evne til at interagere med den verden, vi er omgivet af. Vi kan udføre fysisk hårdt arbejde og løfte tunge ting, men samtidig behersker vi også det finmotoriske som fx at skrive og tegne. Det er denne motoriske spændvidde, der forsøges udnyttet i tangible user interfaces.

I denne artikel vil jeg præsentere tre tangible user interfaces, som alle har det tilfælles, at de betjenes fuldstændig uden mus eller tastatur. Systemerne er valgt fra forskellige retninger inden



Figur 1: Verdens første mus blev udviklet i 1963 af Doug Engelbart og fremvist for offentligheden i 1968.



Figur 2: En landskabsarkitekt er i gang med at designe vha. Illuminating Clay.



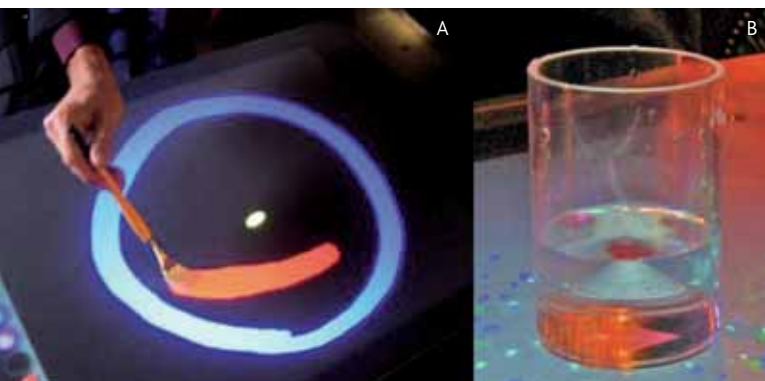
Figur 3: Microsoft Surface var det første kommercielle tabletop tangible interface. Bordet findes i dag rundt omkring på en række kasinoer og luksuriøse hoteller.

for tangible computing og viser tre vidt forskellige måder at benytte fysiske objekter og fysisk interaktion i brugergrænsefladen på. De tre systemer er Illuminating Clay, Microsoft Surface og mixiTUI. Til sidst i artiklen vil jeg, med udgangspunkt i de tre systemer, give et bud på, om musen også i fremtiden vil kunne opretholde samme popularitet, som den har i dag.

Illuminating Clay

Illuminating Clay (tangible.media.mit.edu/projects/illuminatingclay/) er et tangible user interface udviklet på MIT og viser en hidtil uset skulpturel og naturlig interaktionsform. Illuminating Clay er et værktøj udviklet til landskabsarkitekter, landinspektører og ingeniører og benyttes i forbindelse med landskabsplanlægning og nybyggeri. Figur 2 viser en bruger, der interagerer med Illuminating Clay. Det aktive firkantede område, som brugeren holder sin hånd på, er en stor plade af formeligt modellervoks, og det er med dette stykke modellervoks, brugeren betjener systemet. Forvirret? Lad os se på et brugsscenario:

En landskabsarkitekt har fået til opgave at anlægge et nyt parkeringsområde og står derfor nu foran Illuminating Clay-systemet. På bordet foran landskabsarkitekten ses en lille voksmodel af det landskab, hvor parkeringspladsen skal bygges. Ved at trykke let på voksmodellen med fingrene udjævner arkitekten en forhøjning i voksen, så der nu er et fladt område midt i modellen. Idet arkitekten fjerner fingrene fra voksmodellen, skifter farven i det flade område fra gul til rød og advarer derved arkitekten om, at ændringen medfører risiko for jorderosion. En stor vertikal skærm ved siden af arkitekten viser en videoanimation af, hvordan vandet vil flyde i landskabet, og hvor jorden vil blive skyllet bort. Med udgangspunkt i informationerne fra videoen former arkitekten forsigtigt en kanal rundt om det flade område, så vandet kan blive ledt væk fra parkeringspladsen. Farven på voksmodellen ændres følgelig fra rød til gul, og på skærmen viser videoanimationen nu, hvordan vandet strømmer i den nyanlagte kanal.



Figur 4: Microsoft Surface har et højopløseligt genkendelsessystem, som både kan detektere penselstrøg (A) og måle væskeindholdet i glas (B).



Figur 5: En bruger betjener sin mobiltelefon ved hjælp af touch-skærmen på Microsoft Surface.

Sammensmeltning af input og output

Formålet med Illuminating Clay-projektet er at udfordre måden, hvorpå vi tænker digital interaktion. I Illuminating Clay er interaktionen fysisk og organisk og knytter sig tæt til den opgave, den løser. Illuminating Clay er et såkaldt proof of concept-system, og det er således ikke tanken, at systemet skal afløse nuværende produktionssystemer som AutoCAD el.lign.

I Illuminating Clay er afstanden mellem den digitale og den fysiske verden næsten ikke eksisterende – brugeren har helt bogstaveligt fingrene nede i den digitale model. Enhver ændring, der foretages i modellervoksens topologi, fører til en tilsvarende ændring i den digitale model. Dette betegnes inden for tangible computing som *input/output unification* – dvs. sammensmeltning af input og output. I traditionelle grafiske brugergrænseflader er input og output tydeligt adskilt: Computerskærmen udgør systemets output og er brugerens vindue ind til den digitale verden, mens mus og tastatur er brugerens vej til at påvirke den digitale verden og udgør systemets input. I Illuminating Clay interagerer brugeren derimod på samme fysiske "enhed", som han/hun benytter til at observere systemets tilstand med. Det giver derfor ikke mening at klassificere voksmodellen som input *eller* output, men som input *og* output.

Ved at evaluere Illuminating Clay fandt systemdesignerne ud af, at brugerne fik en fornemmelse af rumligheden i det landskab, de arbejdede med. I de systemer brugerne, normalt benyttede, betragtede de det tredimensionelle landskab på en flad skærm, og herved skulle den enkelte bruger selv tolke de rumlige parametre ud fra det betragtede. Brugere fandt det meget givtigt, at de med Illuminating Clay kunne gå rundt om landskabsmodellen, betragte den fra forskellige vinkler og afstande og derved danne sig et bedre billede af landskabet.

Microsoft Surface

I Illuminating Clay består interaktionen i, at brugeren ændrer selve brugergrænsefladens fysiske form. Langt mere normalt er det dog, at interfacets form er statisk, og at det taktile består i, at

brugeren benytter fysiske objekter i interaktionen. Et af de systemer, der er længst fremme med denne type interaktion, er Microsofts Surface-system (www.microsoft.com/surface).

Microsoft Surface tilhører en undergruppe af tangible user interfaces, der betegnes som tabletop tangible interfaces, og som er den hurtigst voksende gruppe af systemer. Betegnelsen dækker over et system, der er designet som et traditionelt bord, men hvor bordpladen udgøres af en touch- og objektfølsom skærm. Som figur 3 viser, er Microsoft Surface designet som et sofabord, og en række af verdens dyreste kasinoer og hoteller har allerede et eksemplar stående i deres lounges.

I tabletop tangible interfaces fanges interaktionen med skærmen vha. et kamera inde i bordet, der affotograferer fysiske objekter eller fingre, der rører skærmen. Det, der adskiller Microsoft Surface fra andre systemer, er, at affotograferingen sker vha. fire kombinerede kameraer. Dette giver et billede med langt højere opløsning end i andre systemer og muliggør en mere detaljeret genkendelse. Eksempelvis kan Microsoft Surface, på grund af den høje opløsning, aflæse stregkoder fra bøger eller dvd'er, der lægges på skærmen.



*Brugeren kan
male direkte
på skærmen*

Hverdagsobjekter som interaktionsenheder

Microsoft Surface benytter det højopløselige genkendelsessystem til at inddrage normale fysiske hverdagsobjekter i den digitale interaktion. Figur 4a viser en maleapplikation, hvor brugeren kan male direkte på skærmen med en ganske almindelig malerpensel. Vha. kameraet detekteres penselstørrelse, vinkel i forhold til skærmen samt information om, hvor hårdt penslen trykkes mod skærmen – alt sammen parametre, som er essentielle for en realistisk tracking af brugerens penselstrøg. Malerapplikationen er først og fremmest udviklet som forskningsprojekt, men Microsoft ser et oplagt potentiale i forhold til undervisning i billedkunst. Her ville applikationen kunne præsentere forskellige malerteknikker for brugeren og træne dem ved at give konstruktiv kritik.

Systemet på figur 4b stammer fra et af Microsoft Researchs mere kommercielle projekter, SurfaceWare, som henvender sig til barer og restauranter, der benytter Microsoft Surface. I SurfaceWare benyttes bordets genkendelsessystem til at estimere, hvor meget væske der er i de glas, der er placeret på skærmen. Informationen om væskemængden sendes fra de enkelte Surface-systemer til en central applikation, som monitoreres af tjenerne på etablisementet. Med hjælp fra systemet kan tjenerne tilbyde gæsten en genopfyldning netop på det tidspunkt, hvor det er mest sandsynligt, at gæsten vil takke ja. Hensigten med systemet er officielt at sikre gæsten en god og opmærksom betjening, men udsigten til mersalg og øget omsætning udgør formentlig også et vist incitament for køberne af systemet.

Microsoft har foreslået udtrykket *implicitte interaktioner* som betegnelse for interaktioner som dem, der er i spil i SurfaceWare. Her foregår interaktionen ikke som en aktiv handling fra brugerens side, men nærmest som en passiv logging af brugerens fysiske interaktioner.

Det digitale knudepunkt

Ud over at forske i, hvordan hverdagsobjekter kan inddrages i den digitale interaktion, forskes der på Microsoft Research i, hvordan Microsoft Surface kan forbedre integrationen og kommunikationen mellem de mange gadgets, vi benytter i vores hverdag. Microsofts vision er, at alle elek-

troniske enheder som digitale kameraer, iPods eller smartphones skal oplyse trådløst om, hvilke funktionaliteter eller services enheden stiller til rådighed for andre enheder. Tanken er derved at lade de forskellige enheder samarbejde og drage nytte af andre enheders hardware og services.

Smartphonens klare fordele er mobilitet og portabilitet – den gør det muligt for os at lytte til musik, surfe på nettet eller se film, mens vi er på farten. Samtidig har smartphonen dog sine klare begrænsninger, og det er eksempelvis nok de færreste, som ville vælge smartphonen som medie, hvis de skal vise vennerne billeder fra ferien eller besvare dagens e-mails. På figur 5 har en bruger derfor lagt sin Samsung BlackJack-smartphone på Microsoft Surface, hvorved smartphonens services vises og stilles til rådighed for brugeren på den store touchskærm. Herved kan brugeren bladde gennem telefonens musiksamling, billeder eller e-mails på en behagelig måde, og hvis et billede skal overføres til en anden enhed, trækker brugeren blot billedet fysisk over på enheden. På denne måde er det Microsofts vision, at Microsoft Surface skal tjene som et digitalt rendez-vous for alle vores gadgets og facilitere kommunikation og overførsler i mellem digitale enheder.

mixiTUI

Det sidste system er musiksystemet mixiTUI (www.mixitui.com), som er udviklet på DIKU. mixiTUI er en tangible sequencer udviklet til at støtte elektroniske musikere, når de skal opføre elektroniske koncerter. Derudover sigter mixiTUI mod at forbedre koncertoplevelsen for publikummet til elektroniske koncerter. Normalt kan den elektroniske koncert være en meget indadvendt koncertoplevelse, hvor musikeren sidder gemt bag en laptop, uden at publikum kan se, hvad der foregår på skærmen. På mixiTUI er alle interaktioner gjort synlige for publikum, så de kan få et indblik i, hvordan musikeren arbejder og skaber den musik, publikum hører.

mixiTUI benytter, ligesom Microsoft Surface, fysiske objekter, men på en anden og mere abstrakt måde. I mixiTUI sammenkædes lyde og effekter med fysiske klodser, så hver lyd eller effekt er kædet sammen med sin egen klods. Modsat de pensler eller glas, der anvendes i Microsoft Surface, har klodserne i mixiTUI ikke nogen fysisk egenskab i sig selv. Klodserne kan betegnes som fysiske links til digital information, og deres funktion er at give det digitale fysisk substans. Dette er en meget almindelig praksis inden for tangible user interfaces, og når de fysiske objekter benyttes på denne måde, omtales de som *tokens*. Ordet token findes ikke på dansk, men betegner fysiske enheder med symbolsk værdi. Souvenirs, poletter, gavekort og stofmærker er alle sammen genstande, der betegnes som tokens på engelsk.

mixiTUIs interaktionsdesign

mixiTUIs tokens er inddelt i tre grupper: looptokens, effekttokens samt en sessiontoken.

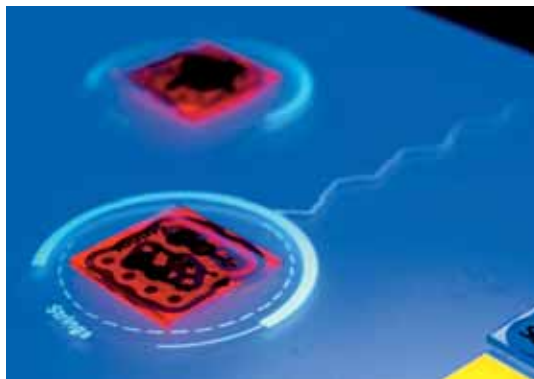
mixiTUIs looptokens udgør den lydgenererende del af mixiTUI og er bundet til digitale loop. Når en looptoken er placeret på skærmen, modtager musikeren feedback om loopets lydstyrke, varighed og aktuelle fremskridt. På figur 7 ses loopets lydstyrke som en bue rundt om tokenen, mens loopets varighed og fremskridt ses som en stiplede linje. Hver streg i den stiplede linje symboliserer et taktslag, og den stiplede linje fungerer på den måde som et ur, der under opførelsen hjælper musikeren med at holde styr på loopenes varighed og fremskridt.



*Klodserne
i mixiTUI
kan betegnes
som fysiske
links*



Figur 6: En elektronisk musiker giver elektronisk koncert på mixiTUI.



Figur 7: En looptoken på mixiTUIs skærm.

mixiTUIs effekttokens er bundet til digitale signalbehandlingseffekter, som fx overdrive-, ekko- eller cuttereffekter. Når en effekttoken er placeret i nærheden af en looptoken eller en anden effekttoken, sammenkobles de to tokens automatisk. Sammenkoblingen mellem tokenerne fører til en ændring i lydens signalvej, som visualiseres ved hjælp af virtuelle ledninger på skærmen. På den måde har musikeren et konstant overblik over, hvilke tokens der er koblet sammen.

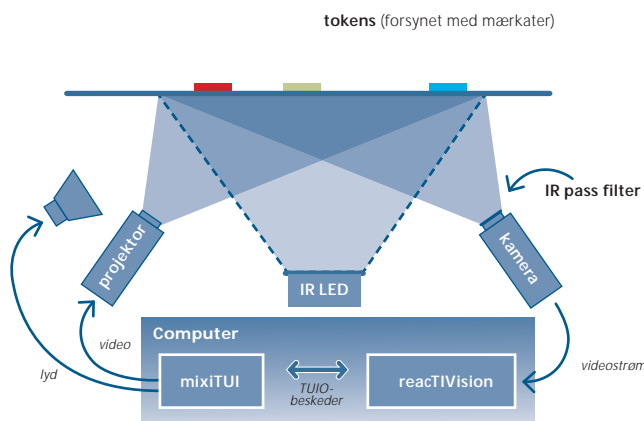
mixiTUIs sessiontoken ses til venstre på figur 6. Musikeren skifter session ved at placere sessiontokenen på en af de fem farver. Når musikeren skifter session, skifter alle looptokens melodi eller rytme, og skærmens baggrundsfarve skifter til den valgte sessionsfarve. På denne måde kan musikeren eksempelvis skifte fra vers til omkvæd uden at skulle tage alle looptokens af og lægge nye på.

Evaluering af mixiTUI

mixiTUI blev evalueret i samarbejde med seks professionelle elektroniske musikere, som alle har stor erfaring med at opføre elektronisk musik live. Musikerne brugte hver især en eftermiddag med mixiTUI, hvor deres egne loop var lagt ind på forskellige looptokens. I det nedenstående vil jeg fremhæve de tilkendegivelser, musikerne kom med i forhold til den fysiske interaktion.

Flere af musikerne fortalte, at de oplevede, at den fysiske tilgang til at mikse musik gjorde, at der var kortere vej fra idé til handling, end når de arbejdede på laptop. En af musikerne udtalte: "Det er intuitivt på den måde, at når jeg får en idé, så kan jeg gøre det med det samme. Alting flytter sig efter det. Hvis jeg rykker på noget, så ændrer det sig."

Ingen af musikerne kendte til begrebet input/output unification, som tidligere er blevet beskrevet. Alligevel hæftede flere af musikerne sig ved, at de på mixiTUI interagerede samme sted, som de observerede, og at dette gav dem en frihed, de ikke var vant til. En af musikerne udtrykte det på denne måde: "Jeg kan godt lide, at det visuelle ligesom træder i kraft. Man kan sætte effekten ind over en lyd, og så er det den, man arbejder med. Så kan man faktisk bare trække den over på en anden, og så er det den, man arbejder med. Det, man ser foran sig, er det, man rent faktisk hører."



Figur 8: Diagram med de enkelte komponenter i mixiTUI.

Samtlige musikere bemærkede, at interaktionen med mixiTUI foregik uden navigation i menuer og vindueskift, hvilket gjorde grænsefladen mere overskuelig. Den fysiske tilgang samt det forbedrede overblik bevirkede, at de var i stand til at improvisere, når de spillede – noget, musikkerne kraftigt efterlyste i eksisterende systemer: "Jeg tror, at det, som elektroniske musikere ofte mangler, det er sgu at lege. Og jeg tror også, at publikum savner, at elektroniske musikere leger."

For at vurdere, om mixiTUI også var i stand til at forbedre publikums koncertoplevelse, blev der afholdt en evalueringskoncert i et auditorium på DIKU, hvor 117 personer deltog. Evalueringen forløb ved, at musikeren "The Mad System" skiftevis opførte elektronisk musik på mixiTUI og på sin laptop. Herefter blev publikum bedt om at udfylde et spørgeskema og forholde sig til en række forskellige parametre, som knyttede sig til koncertoplevelsen. Forsøgsdeltagerne var positive over for mixiTUI, og 103 af forsøgsdeltagerne vurderede, at mixiTUI havde forbedret deres koncertoplevelse. På en skala fra 1 til 7, hvor 1 var dårligst, og 7 var bedst, bedømte forsøgsdeltagerne i gennemsnit mixiTUI-koncerten 1,38 højere end laptop-koncerten.

Den tekniske del

Figur 8 viser et diagram med de enkelte komponenter i mixiTUI.

mixiTUI er et computer vision-system (CV), hvilket vil sige, at de enkelte tokens detekteres vha. visuel genkendelse. Alle tokens har et mærkat påklisteret på undersiden, som bruges i genkendelsesprocessen. Mærkaterne kaldes fiducials og kan på mange måder sammenlignes med de stregkoder, der findes på dagligvarer – blot i en lidt mere avanceret version. Når brikkerne lægges på mixiTUIs skærm, som består af en mat plexiglasplade, kan mærkaterne ses af et kamera, der sidder inde i bordet, og som filmer hele glaspladen. Videostrømmen fra kameraet behandles af et CV-program, reactIVision, der analyserer videostrømmen, søger efter mærkater og bestemmer placeringen og vinklen på de tokens, der er placeret på glaspladen.



Figur 9: Musens fremtid?

Denne information sendes via netværksbeskeder (TUIO-beskeder) videre til mixiTUI-softwaren, hvor den fortolkes i forhold til brugergrænsefladedesignet. Ændringer i placeringen og vinklen på tokens fører således til ændringer i mixiTUI-softwarens tilstand og dermed til ændringer i musikken. Den auditive feedback suppleres med visuel feedback, idet der rundt om de aktive tokens projekteres supplerende grafisk information (figur 7). Herved kan musikeren observere systemets digitale tilstand og aflæse informationer om tokenernes digitale egenskaber, mens der drejes eller flyttes på tokens.

Glaspladen belyses konstant med infrarødt lys, som er usynligt for det menneskelige øje, men synligt for kameraet. Herved sikres det, at kameraet har optimale lysforhold og kan levere et skarpt og kontrastfyldt billede, selv når systemet bruges i (menneskeligt set) mørke omgivelser. For at undgå, at tokendetekteringen generes af det projekterede billede, er der på kameraet påmonteret et IR pass-filter, som bortfiltrerer alt andet end infrarødt lys. Dette bevirker, at billedet fra projektoren ikke fanges af kameraet.

Det er, som tidligere nævnt, det samme princip, der benyttes i Microsoft Surface og andre tabletop tangible interfaces.

Musens fremtid

Styrken ved TUI-systemer som dem, der er blevet præsenteret i denne artikel, er, at systemernes interaktionsform er tæt koblet til den opgave, systemerne løser: Skal der males, foregår det med en pensel, og skal der formes et landskab, foregår det med hænder og modellervoks. Interaktionen virker naturlig og intuitiv for brugeren og kan være svær at efterligne med mus og tastatur. Omvendt kan man sige, at styrken ved traditionelle computere med mus og tastatur er, at der her

netop ikke er en tæt kobling mellem interaktionsform og den opgave, der løses. Det er nemlig denne egenskab, der gør, at den samme computer kan bruges til både tekstbehandling, regneark og browsing. Denne generiske kvalitet besidder de præsenterede TUI-systemer ikke – vil man forme landskaber, må man bruge Illuminating Clay, og vil man spille musik, må man bruge mixiTUI.

De nye interaktionsformer, som ses inden for tangible computing og multi-touch, gør altså ikke med ét musen eller den traditionelle computer forældet. Derimod betyder tilkomsten af de mange nye teknologier blot, at paletten med interaktionsteknikker er blevet bredere, og at interaktionsdesignere ikke mere behøver at vælge mus og tastatur som interaktionsform i mangel af bedre alternativer. Nye avancerede interaktionsteknikker vil løbende blive tilføjet og gøre fremtidens palet endnu mere farverig, men jeg tror fortsat, at musen og tastaturet vil være at finde blandt grundfarverne. ♦

Læs mere

Ishii, H. and Ullmer, B. 1997. Tangible bits: towards seamless interfaces between people, bits and atoms. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (Atlanta, Georgia, United States, March 22 - 27, 1997). S. Pemberton, Ed. CHI '97. ACM, New York, NY, 234-241.

Ullmer, B. and Ishii, H. 2000. Emerging frameworks for tangible user interfaces. IBM Syst. J. 39, 3-4 (Jul. 2000), 915-931.

Underkoffler, J. and Ishii, H. 1999. Urp: a luminous-tangible workbench for urban planning and design. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems: the CHI Is the Limit (Pittsburgh, Pennsylvania, United States, May 15 - 20, 1999). CHI '99. ACM, New York, NY, 386-393.

Mixitui.com

Centrale begreber

Tangible user interface (TUI): En brugergrænseflade hvor interaktionen skabes vha. fysiske objekter.

Tabletop tangible interface (TTI): En underkategori af TUI, hvor fysiske objekter benyttes på en multi-touch-skærm, der er indbygget i et bord.

Multi-touch-skærm: En skærm som kan registrere fingre eller objekter, der berører skærmen. Multi-touch-skærme findes fx på iPhone og iPad.

Token: Et fysisk objekt, der er kædet sammen med digital information eller data. En token kan betegnes som en fysisk repræsentation af digitale data.

Tool: Et fysisk objekt, der er i stand til at manipulere digital information eller data. Et tool er typisk fysisk udformet, så det afspejler sin digitale funktion. Penslen fra Microsoft Surface er et eksempel på et tool.

ESBEN WARMING PEDERSEN



Esben Warming Pedersen er bachelor i musikvidenskab og er kandidat i datalogi fra DIKU. Esben modtog i 2009 SIGCHI.dk's studenterpris for årets mest ekstraordinære speciale. I dag er Esben ph.d-studerende på DIKU, hvor han arbejder med profes-

sionelle elektroniske musikere som brugergruppe. Esbens primære forskningsområder er tangible user interfaces, surface computing og human robot interactions. Esben har publiceret artiklen "mixiTUI: A Tangible Sequencer for Electronic Live Performances" (2009).

Krav til informationssikkerhed

.....
Af **Peter Blume**, Jura, KU
.....

Studerende Christina Jensen går i banken for at hæve et beløb på sin bankkonto. Til sin forfærdelse opdager hun, at kontoen er lænset og står i minus. En anden person har muntret sig med at hæve penge og købe ind i hendes navn – ved brug af hendes CPR- og bankkonto-nummer. Et glemt sygesikringsbevis og bankens sløseri med at kræve fuld legitimation er årsag til hele miseren. Lis har været udsat for identitetstyveri.

Identitetstyveri er blot en af de mange nye trusler mod informationssikkerheden, som er opstået i kølvandet af digitaliseringen i samfundet. Hvor der er mulighed for at skaffe sig lette penge på andres bekostning, vil der altid være personer, der forsøger at snyde systemet. Og har held med det, fordi der ikke findes en 100% sikker metode til at garantere informationssikkerheden.

It- eller informationssikkerhed er et komplekst område, som der ikke bare findes en enkelt, men derimod en række forskellige metoder og tilgange til. Ingen af dem kan stå alene – de må supplere hinanden.

Lovgivningen indeholder regler og retningslinjer, som både private virksomheder og organisationer og offentlige myndigheder med ansvar for egne og andres data skal kende og efterleve. Derudover handler it-sikkerhed især om fornuftig adfærd og om at tage teknikken til hjælp.

Denne artikel giver først en grundig indføring i reglerne ved professor dr.jur. Peter Blume – og suppleres med en faktaboks med en liste over de mest typiske sikkerhedstrusler.

Hvordan kan du være sikker på, at offentlige myndigheder og private virksomheder ikke misbruger oplysninger om dig? Kan du være sikker på, at ingen andre får adgang til oplysningerne? Svaret på disse spørgsmål findes i loven om behandling af personoplysninger, som bl.a. stiller krav til offentlige myndigheders og private virksomheders informationssikkerhed. Artiklen beskriver, hvilke krav der stilles til informationssikkerhed i dag og i fremtiden.

Hvorfor bekymre sig om informationssikkerheden?

I informationssamfundet spiller oplysninger om personer en væsentlig rolle. Det kan fx være oplysninger om kunderne i en virksomhed eller kommunens oplysninger om borgernes indtægter. Sådanne oplysninger kaldes i lovgivningen for personoplysninger.

Når en offentlig myndighed eller en privat virksomhed opretter et register med personoplysninger, kan de bruge oplysningerne til mange nyttige formål, som sikrer, at borgerne og kunderne får en god, hurtig og effektiv behandling. Der er dog altid en risiko for, at personoplysningerne kan bruges til formål, som den enkelte person ikke ønsker, eller som er ulovlige. Eller endnu værre: at personer uden for myndigheden eller virksomheden kan få adgang til fortrolige personoplys-

ninger. For at sikre borgerne mod misbrug af persondata har man i Danmark og i en række andre lande vedtaget regler om databeskyttelse og informationssikkerhed. I Danmark står reglerne i ”lov om behandling af persondata”, også kendt som ”persondataloven”.

Hovedformålet med reglerne er at undgå, at personoplysninger misbruges – med den konsekvens, at den enkelte persons integritet og privatliv krænkes. Risikoen herfor kan opstå i et utal af situationer. Som eksempler kan nævnes misbrug af CPR-numre og passwords med det formål at læse bankkonti, hacking af hjemmesider eller udstilling af personers portrætter på hjemmesider.

Informationssikkerhed er en fundamental del af informationssamfundet, fordi reglerne om sikkerhed har stor betydning for den tillid, som informationsteknologien mødes med. Spørgsmålet er bare, hvordan man sikrer, at reglerne fungerer efter hensigten, således at der er tale om en reel sikkerhed. Som hovedregel skal det sikres, at personoplysninger kun anvendes af autoriserede personer til autoriserede formål. Dette er en forudsætning for, at borgerne kan have ikke blot tillid til systemerne, men også tryghed, fordi brugen af teknologien bliver mere gennemskuelig for den enkelte.



Hvad siger persondataloven om informationssikkerhed?

Persondataloven har en regel om informationssikkerhed i § 41, stk. 3, som fastslår, at "Den dataansvarlige skal træffe de fornødne tekniske og organisatoriske sikkerhedsforanstaltninger mod, at oplysninger hændeligt eller ulovligt tilintetgøres, fortabes eller forringes, samt mod, at de kommer til uvedkommendes kendskab, misbruges eller i øvrigt behandles i strid med loven. Tilsvarende gælder for databehandlere."

Persondataloven indeholder de grundlæggende regler om de sikkerhedskrav, der skal være opfyldt, når man anvender personoplysninger. Selvom der også i andre sammenhænge gælder regler om sikkerhed, er det først og fremmest i forhold til personoplysninger, at sådanne krav er aktuelle, og det er derfor beskyttelsen af personoplysninger, der er temaet i denne artikel. Det er de skadevirkninger for borgerne, som manglende eller utilstrækkelig informationssikkerhed kan medføre, der især påkalder sig opmærksomhed.

I forhold til personoplysninger kan to situationer sikkerhedsmæssigt forekomme. For det første kan den, der råder over oplysningerne (*den dataansvarlige*), selv opbevare dem og foretage dispositioner, og for det andet kan opbevaringen være overladt til nogen, der har bedre eller større teknisk kapacitet (*en databehandler*). Persondatalovens regler tager sigte på begge situationer, idet det først og fremmest er den dataansvarlige, der har ansvaret for, at sikkerheden er i orden.

Efter persondataloven skal den dataansvarlige "træffe de fornødne tekniske og organisatoriske sikkerhedsforanstaltninger mod, at oplysninger hændeligt eller ulovligt tilintetgøres, fortabes eller forringes, samt mod, at de kommer til uvedkommendes kendskab, misbruges

eller i øvrigt behandles i strid med loven". Det står også i loven, at personer, der arbejder for en dataansvarlig eller en databehandler, skal følge en instruks fra den dataansvarlige. Hvis den dataansvarlige benytter en databehandler, skal der være en skriftlig aftale mellem den dataansvarlige og databehandleren. Loven fortæller os således, at der skal være sikkerhed, og hvem der har ansvaret herfor, idet det også fremgår, at det er strafbart ikke at leve op til disse krav.

Loven fortæller os derimod ikke, hvilke mere præcise sikkerhedsforanstaltninger der skal være sat i værk, og ej heller om man skal bruge de samme foranstaltninger i alle situationer. Denne tavshed illustrerer en vigtig forskel på den juridiske og tekniske regulering af informationssikkerhed. Det er en almindelig iagttagelse, at de tekniske krav, der skal opfyldes for at opnå acceptabel sikkerhed, ofte er komplekse, og at de ikke er stationære, fordi de hele tiden må udvikles og tilpasses til nye former for teknologi. Kravene er i det hele taget betinget af, hvilken teknologi der bliver anvendt. Der er eksempelvis forskel på den situation, hvor personoplysninger opbevares i et centralt mainframe-system, og den situation, hvor oplysningerne er placeret i et netværk. Der er således mange forskellige momenter, som må tages i betragtning ved udformning af sikkerhedskrav. Det er endvidere karakteristisk, at mange sikkerhedskrav er teknisk formuleret, dels for at de kan være tilstrækkeligt præcise, og dels fordi de typisk skal realiseres af personer, der besidder teknisk sagkundskab.

Forhold af denne karakter fører til, at alle de forskellige detaljerede krav til sikkerhed, der skal opfyldes, ikke er et emne, som er egnet til at blive medtaget i en lov. For det første ville loven i så fald blive meget omfangsrig, og for det andet ville det være nødvendigt hele tiden at ændre loven, hvilket ikke er

realistisk. En lov skal altid behandles flere gange af politikerne i Folketinget, og det tager ofte lang tid. Man må derfor bruge andre reguleringsformer, der bl.a. har den fordel, at de giver en vis fleksibilitet. Derfor har Folketinget bedt justitsministeren om at skrive de detaljerede regler, som skal bruges i de offentlige myndigheder. Justitsministeren kan hurtigt skrive reglerne om, hvis der fx kommer nye tekniske sikkerhedsløsninger eller nye behov. Folketinget har også bedt Datatilsynet om at hjælpe myndighederne med at overholde reglerne og om at holde øje med, om reglerne overholdes. Derfor har Datatilsynet skrevet en vejledning om reglerne om informationssikkerhed. Justitsministeren har ikke skrevet regler til de private virksomheder eller private organisationer om sikkerhed for personoplysningerne, hvilket ofte give anledning til en vis undren. Årsagen er formodentlig, at man ikke vil regulere den private sektor for tæt, samt at denne sektor er mere forskelligartet end den offentlige sektor. I praksis mener man dog, at private virksomheder som udgangspunkt skal følge de regler, der gælder for de offentlige myndigheder.

Hvordan skal en myndighed eller virksomhed opbevare data?

Ansvar for at opbevare personoplysninger sikkert og forsvarligt ligger først og fremmest hos den myndighed eller virksomhed, der råder over oplysningerne (den dataansvarlige). Denne kan vælge selv at opbevare data og foretage dispositioner, eller opbevaringen kan overlades til nogen, der har bedre eller større teknisk kapacitet (en databehandler).

Der er ikke faste regler for, hvordan it-sikkerheden i en virksomhed eller myndighed skal gribes an i praksis. Det varierer – alt afhængigt af virksomhedens størrelse, hvor følsomme data der arbejdes med, og om der er adgang for eksterne til at logge sig på netværket. Helt overordnet skal der både internt og eksternt etableres det, som loven kalder ”den fornødne sikkerhed”.

De ansatte skal udvise en fornuftig adfærd

Arbejdsgivere skal fastlægge nogle rutiner eller indføre nogle systemer, som medfører, at de ansatte kun har adgang til de oplysninger, som de har ret til at anvende. Desuden skal arbejdsgiveren kontrollere, at de ansatte kun bruger personoplysningerne til de formål, som de må bruge dem til. De ansatte skal i det hele taget uddannes til at tænke på it-sikkerhed i deres omgang med computere og netværk. Dette kan typisk gøres ved kurser og interne kampagner, der fortæller om god it-adfærd.

Systemerne kan sikres på mange måder

Rent teknisk skal virksomheden sikre sig mod, at personer udefra kan trænge ind til personoplysningerne og herefter bruge, manipulere eller stjæle dem. Typiske foranstaltninger er firewalls og antivirusprogrammer. Internettet er en nyttig kilde til information, men også genstand for talrige målrettede forsøg på ulovlig indtrængen. En virksomhed eller myndighed skal som minimum sørge for, at der ikke ligger personoplysninger åbent tilgængelige på internettet, der kan opsnappes og bruges af andre.

Er alt dette vellykket, hvad det langt fra altid er, er der skabt en større sandsynlighed for, at borgerne opnår den databeskyttelse, som lovgivningen tilsigter.

Virksomheder og myndigheder har en stor interesse i at beskytte deres oplysninger, fordi de repræsenterer en betydelig værdi. Herudover bidrager de teknologiske sikkerhedsløsninger også til at forebygge kriminalitet og undgå andre sikkerhedstrusler som følge af fx hacking. Undertiden udtrykkes dette, som at ”svaret på teknologien er teknologien”.

Kravene til it-sikkerhed skal være realistiske

Kommet så langt er spørgsmålet, hvilke krav virksomheden eller myndigheden skal stille til sin informationssikkerhed. I første omgang er dette ikke et teknisk spørgsmål, men snarere et samfundsmæssigt. Spørgsmålet opstår først og fremmest, fordi sikkerhed ikke er gratis. Der må tages stilling til, hvor store ressourcer der skal anvendes til dette formål. Spørgsmålet kan også vendes om: Hvilken pris skal virksomheden eller myndigheden betale for adgangen til at behandle personoplysninger?

Når dette spørgsmål er relevant, skyldes det, at personoplysninger er informationssamfundets primære grundstof, og at der derfor er en almindelig samfundsmæssig interesse i, at der er adgang til at benytte disse oplysninger. Nogle sikkerhedskrav er som udgangspunkt uønskelige, hvis de er så omkostningskrævende, at de reelt hindrer myndigheder eller virksomheder i at benytte personoplysninger på en teknologisk fornuftig måde. Kravene må fastlægges under hensyntagen til de generelle samfundsmæssige behov.

”

En dataansvarlig skal som minimum sørge for, at der ikke ligger personoplysninger åbent tilgængelige på internettet



Megen
informations-
teknologi
er svær at
forstå

Denne indfaldsvinkel fører til, at sikkerhedskravene skal være realistiske. De kan også være forskellige, alt efter hvilke personoplysninger der anvendes, og måden, det sker på. Kravet er, at der skal være en *tilstrækkelig* sikkerhed. Det er altså ikke et krav, at der skal anvendes den nyeste eller absolut bedste sikkerhedsløsning (state of the art). Herudover afhænger sikkerhedsniveauet af andre forhold. Man kan eksempelvis se på, om der behandles følsomme eller almindelige personoplysninger, om formålet med behandlingen kan være risikabelt for personernes integritet, og om oplysningerne befinder sig i eller transmitteres i netværk. Alt i alt skal sikkerheden ikke være optimal, men den skal være *tilstrækkelig* til, at der foreligger en reel databeskyttelse.

Skal sikkerhedskravene forstås af alle – eller bare virke?

Sikkerhedsforanstaltninger kan være komplekse eller simple, idet begge typer kan give anledning til vanskeligheder. Der kan være tale om løsninger, hvis anvendelse enhver kan forstå, og løsninger, der alene kan gennemskues af personer, som har høj teknologisk indsigt. Nogle løsninger er lette at anvende, mens andre er eller opleves som besværlige (mere om dette under 5).

Som udgangspunkt er det ikke et krav, at sikkerhedsforanstaltningen skal kunne forstås af enhver. Den skal blot virke og kunne leve op til sit formål. I færdselslovgivningen stilles der tilsvarende en række krav til biler, og de skal – og bliver det for det meste – overholdes, uden at det er et krav, at den enkelte bilist forstår, hvordan de virker. Afgørende er funktionen. Det gælder også i forhold til informationsteknologien.

Foranstaltningen skal gøre teknologien sikker, således at personoplysningerne befinder sig i et trygt miljø. Selvom reglerne ikke kræver, at de, som anvender personoplysninger, har teknologisk viden, kræver de, dels at der er sikkerhedsløsninger til rådighed, og dels at de anvendes i passende omfang. Begge dele forudsætter, at der er en vis forståelse af de risici, som brugen af teknologien repræsenterer. Denne forudsætning er ikke altid opfyldt, fordi megen informationsteknologi er svær at forstå. Der er mange eksempler på, at truslerne ikke tages alvorligt. Nogle gange ser man også eksempler på, at sikkerhedsforanstaltningerne overdrives, fordi leverandørerne har en økonomisk interesse i at sælge sikkerhedsløsningerne.

Nogle illustrerende eksempler på sikkerhedsforanstaltninger

Den vel nok mest kendte sikkerhedsforanstaltning er *passwordet*. Den person, der er berettiget til at anvende en bestemt computer til bestemte formål, skal anvende en personlig kode for at kunne bruge computeren. Uanset at dette er en særdeles simpel foranstaltning, der tilsigter at holde andre ude, er den central i forhold til informationssikkerheden, fordi den giver mulighed for at identificere brugeren og om fornødent registrere (logge) anvendelsen af personoplysninger. Så er der større sandsynlighed for, at misbrug opdages. Alt dette er dog ikke ensbetydende med, at brugen af passwords er en selvfølge, eller at de anvendes på en forsvarlig måde. Det er nødvendigt at huske passwordet, hvilket gør brugen af personoplysninger mere besværlig.

I en af de mest kendte sager, som Datatilsynet har haft i nyere tid, måtte tilsynet anvende et års tid på at få politiet til at anvende passwords på de enkelte stationer, hvor computerne hidtil havde stået åbne, således at enhver i lokalet frit kunne anvende dem. I forhold til en dataansvarlig, der behandler meget private personoplysninger, gav selv en simpel sikkerhedsforanstaltning anledning til vanskeligheder. Reglerne om sikkerhed er klare og lette at forstå, men det er ikke ensbetydende med, at praksis er i overensstemmelse hermed. Passwords skal behandles fortroligt, men der er sikkert fortsat en del tilfælde, hvor de er skrevet på en lap papir ved siden af computeren



eller på tilsvarende måde ikke holdt fortrolige. Den menneskelige faktor vil altid udgøre et svagt led i sikkerheden.

Der spørges ofte om, hvorvidt sikkerhedsfejl kan afdækkes. Svaret herpå er, at det kan de ikke fuldt ud, men Datatilsynet råder over visse muligheder for at opdage dem. Myndigheder og virksomheder, som registrerer og behandler persondata, skal altid give Datatilsynet besked, når de registrerer og bruger personoplysninger. Det sker ved, at den dataansvarlige myndighed eller virksomhed sender en anmeldelse til Datatilsynet, hvor de beskriver de personoplysninger, de registrerer, og hvad de bruger dem til. Datatilsynet kan så gennemføre inspektioner hos de offentlige myndigheder og private virksomheder og organisationer, som registrerer og bruger personoplysninger. Disse inspektioner kræver mange ressourcer og gennemføres derfor kun i begrænset omfang. De har ikke alene til formål at gennemføre kontrol, men danner ligeledes rammen om en dialog mellem virksomheden eller myndigheden og Datatilsynet. Her kan man bl.a. drøfte, hvorledes sikkerheden bedst tilrettelægges i den konkrete myndighed eller virksomhed. Når det er muligt, kan man finde frem til pragmatiske løsninger.

Et andet spørgsmål er, hvilket sikkerhedsniveau man overhovedet kan kræve. Dette afhænger af den risiko, som brugen af en bestemt teknologi i forhold til bestemte typer af personoplysninger indebærer. Svaret på spørgsmålet kræver indsigt i både jura og teknik.

Et klassisk spørgsmål drejer sig om, hvordan man bedst beskytter e-post. Set med juridiske øjne er en e-post et brev ("lukket meddelelse"). Læser andre personer en e-post uden tilladelse, sker der et brud på brevhemmeligheden. Spørgsmålet er nu, om en e-post faktisk svarer til et brev, eller om den snarere kan sammenlignes med et åbent postkort. Ved udformningen af kravene i lovgivningen har man sidestillet e-post med et åbent postkort. Det har haft den konsekvens, at fortrolige eller følsomme personoplysninger ikke kan sendes med e-post, medmindre de er beskyttet. Hovedmidlet har været *kryptering*, der netop værner kommunikationsindholdet, men ikke de kommunikerendes identitet. Som det vil fremgå nedenfor, giver kryptering dog også

Krypteringsproblemet belyst ved et eksempel

Problemstillingen har været aktuel i forbindelse med en sag om reservationer over internettet af bøger fra biblioteker. Når der reserveres bøger ved brug af e-post, resulterer det i en tilsvarende kvittering pr. e-post fra biblioteksdatabase. Da det er fortroligt, hvilke bøger en borger låner på biblioteker, skal denne kvittering krypteres, hvilket i praksis betyder, at et sådant system ikke fungerer.

Den aktuelle sag endte med, at biblioteksvæsenet fik en dispensation, der oprindeligt var tidsbegrænset, men i foråret 2010 blev forlænget indtil videre. Dispensationens baggrund er, at borgeren efter Datatilsynets opfattelse ikke skal kunne give samtykke til at modtage ukrypteret post. Dette vil svare til at kræve et samtykke til en reduceret sikkerhed. Når det tages i betragtning, hvor ressourcekrævende sikkerhed og kryptering er, ville ordningen formentlig resultere i, at mange flere dataansvarlige ville forsøge at opnå et sådant samtykke, hvorved databeskyttelsen på denne måde blev undergravet. Sagen har udløst stor debat og illustrerer et væsentligt problem i forbindelse med sikkerhed.

anledning til problemer. Selvom krypteringskravet fortsat opretholdes i vidt omfang, er holdningen blevet en anelse mere afslappet. Det skyldes måske manglende kendskab til teknologien, når e-posten uden forbehold sammenlignes med postkortet. Det kræver således en vis teknologisk viden at skaffe sig adgang til andres e-post. Den er vel ikke sikret, men den er heller ikke åben for enhver.

Selvom kryptering i og for sig er en enkel løsning, er denne metode ikke problemfri, fordi det ikke er alle, der er i stand til at læse en krypteret e-post. Faktisk er det de færreste, der er i stand til det. Det har derfor været diskuteret, om en person skulle gives adgang til at sige ja til at modtage ikke-krypteret e-post, selvom den indeholder følsomme eller fortrolige oplysninger.

Sikkerhedsløsninger skal være brugervenlige

Udgangspunktet for reglerne i persondataloven er, at den moderne informationsteknologi er nyttig, og at det er vigtigt for samfundet, at den anvendes. Men det skal ske på en civiliseret måde. Teknologien skal kunne anvendes efter hensigten – og det er samtidig ikke meningen, at sikkerhedsløsningerne skal forhindre anvendelsen. Derfor bør sikkerhedsløsningerne være brugervenlige. Brugen af personoplysninger er så udbredt – også i privatsfæren – at det i mange tilfælde er den enkelte person, der selv må beskytte sine oplysninger. Ikke mindst på internettet er personen ofte alene. Som det omtales nærmere nedenfor, burde sikkerheden ideelt fungere helt uafhængigt af brugeren, men dette er langt fra altid tilfældet.

”
*Sikkerheds-
løsningerne
skal ikke
forhindre
anvendelsen*

Et velkendt eksempel på de problemer, som manglende brugervenlighed kan medføre, er anvendelsen af den *digitale signatur*. Denne beskyttelsesforanstaltning bruges til at dokumentere, hvem brugeren er, hvilket er nødvendigt i den virtuelle verden, hvor vi ikke mødes fysisk, fx i kontakten mellem borger og myndighed via selvbetjeningsfunktionen på en hjemmeside. Signaturen har dog vist sig alt for svær at anvende. Det er en udbredt erfaring, at en internettjeneste ikke bruges i det tilsigtede omfang, hvis det er en betingelse, at der skal benyttes digital signatur. Dette er naturligvis ikke hensigtsmæssigt, og det er derfor en væsentlig ambition bag reglerne om informationssikkerhed, at det tilstrækkelige sikkerhedsniveau ikke blot er til stede, men samtidig har den fornødne brugervenlighed. Som allerede nævnt er dette ikke nogen nem målsætning at realisere.

Virksomheder og myndigheder skal forberede sig på sikkerhedstruslerne

Sikkerhed kommer som bekendt ikke af sig selv. Det er vanskeligt for en del dataansvarlige at vælge den teknologi, som bedst sikrer persondatabeskyttelsen i netop deres virksomhed eller myndighed. Det tager tid, og det kan være fristende at lade være med at afsætte de nødvendige ressourcer. For at modvirke dette fænomen er det en fremtrædende tendens i lovgivningen, at der stilles mere systematiske krav til den dataansvarlige.

Er der eksempelvis tale om et større informationsteknologisk system som fx et komplekst økonomistyringssystem med fortrolige data, mange forskellige brugerroller eller flere separate registre, der ikke må samkøres, stiller Datatilsynet i dag ofte krav om, at der i forvejen skal være gennemført en analyse af de konsekvenser for databeskyttelsen, som det pågældende system har. Infrastrukturen skal være kortlagt, og der skal være foretaget en såkaldt PIA (privacy impact assessment, bedst oversat som vurdering af konsekvenser for persondatabeskyttelsen), hvori sikkerhed indgår som et vigtigt forhold. Virksomheden eller myndigheden skal således være på forkant med de udfordringer, som en bestemt anvendelse af personoplysninger har, således at det på forhånd kan udelukkes, at misbrug af personoplysninger vil forekomme.

Når det gælder selve udformningen af teknologien, er det også karakteristisk, at der ikke findes regler for teknologien som sådan, fordi teknologien hele tiden udvikler sig, og man kan derfor kun vælge den teknologi, som lever op til de gældende sikkerhedskrav.

En lovmæssig regulering indebærer en risiko for, at kravene alligevel ikke opfyldes, fordi det ikke er sikkert, at den dataansvarlige kan administrere de frihedsgrader, der er fastlagt. På denne baggrund er der nu introduceret et nyt begreb, *privacy by design*, som betyder, at den nødvendige sikkerhed skal være indbygget i den teknologiske løsning på forhånd.

Et eksempel kan være, at persondata krypteres og dekrypteres, uden at brugeren selv behøver at gøre noget for, at dette sker. Den teknologi, der benyttes til at behandle personoplysninger, skal generelt være beskyttende (*privacy enhancing technology*), bl.a. ved at tilvejebringe en god sikkerhed. Dette betyder ikke nødvendigvis, at teknologien er brugervenlig, selvom dette også er en forventning. Det øger dog sandsynligheden for, at der ikke sker misbrug af personoplysningerne.

Privacy by design, der ikke i skrivende stund er indarbejdet i persondataloven, behøver ikke i sig selv at få en positiv betydning. Det jo ikke er sikkert, at den dataansvarlige faktisk benytter disse former for teknologi. Der skal derfor endnu mere til. Dette "mere" kunne være, at der gennemføres en certificeringsordning for den teknologi, der må benyttes til at behandle persondata, i hvert fald i større sammenhænge. Med en certificeringsordning vil en bestemt teknologi være udpeget til at være privatlivsbeskyttende. Denne teknologi, der kan være en blandt flere anvendelige muligheder, vil kunne benyttes af den dataansvarlige i forhold til bestemte former for behandling af personoplysninger. Det er på mange måder et ambitiøst og krævende projekt, der er udtryk for den opfattelse, at sikkerhedsniveauet i dag (2010) generelt bedømt ikke er tilfredsstillende.

Vedtages der en certificeringsordning, bliver det en blandet juridisk og teknisk opgave at foretage de vurderinger, der er nødvendige for valget af egnede teknologier. Som udgangspunkt fastlægger juristerne de krav, der skal opfyldes, hvorefter det teknisk kan vurderes, om en bestemt teknologi lever op til disse krav. Der er således tale om en tværfaglig opgave, hvor det er væsentligt, at jurister og teknisk kyndige forstår hinanden. Denne opgave er særlig krævende, fordi personoplysninger anvendes internationalt. Skal en certificering have nogen mening, må den i princippet gælde overalt. Den konstante transmission af personoplysninger via internettet illustrerer, at rent nationale løsninger har begrænset værdi og endda skader konkurrenceevnen for de dataansvarlige, som forpligtes af dem. Det er derfor sandsynligt, at hvis denne ordning gennemføres, bliver der i det mindste tale om en form for EU-certificering.

Sikkerhedsbrister opdages ofte rent tilfældigt

I disse år er der en fornyet fokus på behovet for sikkerhed. I vidt omfang er det en tværfaglig opgave at realisere den sikkerhed, der er påkrævet for at sikre personoplysninger imod misbrug. Uanset de mange bestræbelser er det dog ikke realistisk at forestille sig, at der aldrig sker sikkerhedsbrud. Spørgsmålet er, dels hvordan man bliver klar over, at der er sket et sådant brud, dels hvilke konsekvenser det har.

Nogle sikkerhedsbrud afdækkes nærmest af sig selv, fordi de har så store konsekvenser, at de uden videre opdages, eksempelvis tyveri af pc'er med fortrolige data eller hacking, der medfører

Gode råd om it-sikkerhed

IT- og Telestyrelsens hjemmeside giver med domænet it-borger.dk/sikkerhed et glimrende overblik over junglen af it-sikkerhedstrusler samt midler til at imødegå dem. Eksemplerne nedenfor på typiske sikkerhedstrusler anno 2010 er således hentet fra denne side, som også giver en udtømmende oversigt over tekniske og adfærdsmæssige forholdsregler.

SPYWARE

Spyware er små spionprogrammer, som – uden din viden – indsamler informationer om dine interesser og vaner på internettet. Disse informationer sender spywaren tilbage til bagmændene, som enten selv bruger oplysningerne i reklameøjemed eller sælger oplysningerne videre til andre.

Cookies

Cookies anvendes til at gemme oplysninger om dig på din egen computer. Cookies er som regel ikke skjult for brugerne, men du kan forhindre, at der gemmes cookies på din computer ved at ændre i din browsers internetindstillinger under fanebladet sikkerhed.

Adware

Adware-programmer er mere harmløse end spyware, da de ikke sender information tilbage til bagmændene uden din viden. Adware kan vise pop op-vinduer med reklamer, ændre dine indstillinger i browseren (for eksempel startside), installere uønskede søgemenuer eller tilføje bookmarks i browseren.

Browserhijack

Browserhijacking (kidnapning af din browser) forveksles tit med spyware. Browserhijacking er, hvis du bliver ført til en anden hjemmeside end forventet, selvom du har indtastet den korrekte webadresse.

Keyloggere

En keylogger er en form for spyware, som er et lille program, som registrerer, hvilke

taster du trykker på, og herefter sender det til bagmanden. Dette kan for eksempel bruges til at opfange passwords eller anden fortrolig information.

Malware

Malware er en sammenskrivning af "malicious software", som betyder ondsindede programmer. Malware er betegnelsen for ethvert program, som er ondsindet over for en bruger. Malware dækker både over vira, orme, trojanske heste, keyloggere, spyware og lignende.

VIRUS

En computervirus er et program, der er i stand til at sprede sig til andre computere. Virussen kan ligge i et program eller en fil og udføre skadelige aktiviteter på din computer, fx slette filer eller programmer.

Orm

En orm er ikke knyttet til et program eller en fil. Ormen spreder sig selv over det netværk, som computeren er tilkoblet. Det kan både være det lokale netværk eller internettet. Da ormen er afhængig af en forbindelse over et netværk, er servere de mest udsatte, fordi de som regel har fast forbindelse til nettet.

Trojansk hest/bagdør

En trojansk hest er et program, der giver en "hemmelig" adgang til din computer. Den trojanske hest kan fx være skjult i et spil eller program, du har downloadet fra en hjemmeside, eller din computer kan være inficeret af en virus, der selv har installeret den trojanske hest. Hackere eller virusbagmændene kan bruge den trojanske hest til at få adgang til din computer, og dermed får de mulighed for at overtage kontrollen med din computer, herunder finde fortrolige oplysninger, gemme ulovlige programmer eller bruge din computer til at sende spam til andre.

Hoax

Hoax betyder spøg eller svindelnummer. En hoax er altså ikke en egentlig virus

eller et program, men en falsk advarsel i en e-mail om en virus, der ikke findes. En hoax spredes ved, at folk i god tro sender advarslen videre. En hoax kan fx have en tekst, hvor der står noget i retning af: "Send straks denne advarsel videre til alle dine venner".

PHISHING – IDENTITETSTYVERI

Identitetstyveri sker, når personer tilegner sig andres personoplysninger og udgiver sig for at være disse personer. Det kan ske elektronisk ved brug af bankoplysninger, CPR-numre eller kodeord eller ved at bruge den andens id-papirer (sygesikringsbevis, kørekort m.m.). Der er også tale om identitetstyveri, når en person køber produkter, fx over internettet, ved hjælp af en andens person- og kontooplysninger. Phishing foregår, hvis man modtager en falsk e-mail, der ser ud til at komme fra en bank, onlinebutik, forsikringselskab eller en anden troværdig afsender. I e-mailen opfordres man til at opgive personoplysninger, numre på betalings- og kreditkort, kontonumre, CPR-numre, pinkoder eller lignende og derpå sende e-mailen tilbage med oplysningerne.

MOBILSVINDEL VIA "ONE-RING FRAUD"

Som mobilbruger får du ofte en sms om, at en person forgæves har forsøgt at få kontakt med dig. Men ringer du op på det nummer, der bliver vist, kan du i særlige tilfælde komme til at betale dyrt.

FUPHJEMMESIDER

Staver du forkert, når du taster en netadresse ind i din browser, kan du risikere at havne på en hjemmeside, der indeholder skadelige programmer. Skriver du fx www.qZl.dk i stedet for www.qXl.dk eller www.kraRk.dk i stedet for www.krak.dk, kan du lande på en hjemmeside, der er designet til svindel.

Kilde: www.it-borger.dk/sikkerhed

systemnedbrud eller afsætter klare visuelle aftryk. I andre tilfælde er det mindre klart, om en uhensigtsmæssig anvendelse af personoplysninger skyldes et sikkerhedsbrud, og ligeledes uklart, hvor og hvornår dette er sket. Det kan være, hvis personer i en virksomhed eller myndighed bevidst eller ubevidst udnytter deres adgang til persondata på en uhensigtsmæssig eller uautoriseret måde. Datatilsynet har begrænsede ressourcer, og uanset adgangen til at foretage inspektioner er det ikke sandsynligt, at tilsynet bliver opmærksomt på alle de sikkerhedsbrud, der rent faktisk forekommer. Det er ligeledes usandsynligt, at borgerne altid opdager disse brud.

På denne baggrund er der indført et nyt middel, der i første omgang kun gælder for elektroniske kommunikationstjenester, men som forventes at komme til at gælde generelt, nemlig en meddelelsespligt. Den dataansvarlige har således nu pligt til at orientere Datatilsynet om sikkerhedsbrud, der har haft en praktisk betydning. Hvis sikkerhedsbruddene også indebærer en risiko for integritetskrænkelser, skal de berørte borgere informeres. Denne meddelelsespligt ("*breach notification*") skal bl.a. skabe større åbenhed om sikkerhedsniveauet. Det forudsætter dog, at den dataansvarlige faktisk giver denne meddelelse. Selvom dette næppe vil ske i alle tilfælde, er det sandsynligt, at de myndigheder og virksomheder, der er ansvarlige for mange personoplysninger, vil respektere denne ordning, som dermed sandsynligvis vil bidrage til at skabe en bedre sikkerhed og dermed større tryghed.

Når det konstateres, at der er sket et sikkerhedsbrud, kan retssystemet, dvs. politiet og domstolene, reagere på forskellige måder. Den dataansvarlige kan blive straffet for at overtræde sikkerhedsreglerne, men det er ganske sjældent, at nogen bliver straffet for sikkerhedsbrud. I princippet kan den krænkede person kræve erstatning, hvis den dataansvarlige kunne have gjort noget for at undgå, at den registrerede led et økonomisk tab. Men sådanne erstatningskrav fremsættes stort set aldrig. Det er kun i forhold til få typer af persondatabehandling, at et sikkerhedsbrud kan få den konsekvens, at den dataansvarlige fratages retten til at behandle persondata. Det ville forudsætte, at der gentagne gange er forekommet store sikkerhedsbrud. Reglen er endnu aldrig blevet brugt, og det er som nævnt også sjældent, at der sker andre reaktioner. I forhold til reaktionsmidler kan man godt konkludere, at retssystemets præstation ikke er imponerende.

Datasikkerhed – den endeløse saga

Den gode sikkerhed opstår nok først og fremmest, fordi det er i den dataansvarliges egen interesse at fremme den, eller fordi den dårlige omtale, som sikkerhedsbrud kan føre til, er både økonomisk og image-mæssigt skadelig for den dataansvarliges virksomhed. Selvom der hele tiden vedtages nye regler, der tager sigte på at fremme god informationssikkerhed, og selvom der samfundsmæssigt er stor fokus på dette område, vil emnet også fremover være højt profileret.

Der har altid været og vil fremover blive ved med at være god og dårlig sikkerhed. Der vil også fremover være en kamp mellem de teknologier, der understøtter databeskyttelse, og teknologier, der undergraver denne beskyttelse. Denne kamp slutter næppe nogensinde, og den er særlig fremtrædende i disse år, hvor nye teknologier og nye måder at anvende kendt teknologi på konstant viser sig. Teknologien og juraen må stå side om side i denne kamp. Målet og håbet må være, at de fleste sejre vindes af de teknologier, der skaber den gode sikkerhed og dermed den effektive beskyttelse af borgernes personlige integritet og privathed. ♦

Læs mere

Peter Blume: *Databeskyttelsesret* (3.udg. 2008 DJØF Forlag)

Peter Blume, Janne Rothmar Herrmann: *Ret, privatliv og teknologi* (2.udg. 2010 DJØF Forlag)

PETER BLUME



Peter Blume har siden 1993 været professor i retsinformatik på Det Juridiske Fakultet på Københavns Universitet. Peter Blume blev cand.jur. i 1974, lic.jur. i 1982 og dr.jur. 1989 og var institutleder på Retsvidenskabeligt Institut i perioden 1982-2002.

Han er og har været medlem af adskillige råd og udvalg, herunder Datarådet, Akademisk Råd, Registerlovsudvalget samt formand for TV-overvågningsudvalget i 2006.

I dag er Peter Blumes primære forskningsområde persondataret samt juridisk metode, som han også underviser i på Københavns Universitet. Af udvalgte publikationer kan nævnes: "Databeskyttelsesret (2008)" og "Persondataretten – nu og i fremtiden" (2010).

Digitale medier

.....
Af **Rasmus Helles**, MEF, KU
.....

Udviklingen af de digitale teknologier har muliggjort udviklingen af en lang række nye medier, hvoraf flere på rekordtid er blevet taget i anvendelse af store dele af befolkningen. Artiklen beskæftiger sig med forholdet mellem opfindelsen af de digitale teknologier i sig selv og den tilsvarende opfindelse af deres kulturelle og sociale anvendelser som medier.

Medier og kommunikation

Ordet ”kommunikation” bruges ofte som en selvfølgelig betegnelse for den centrale menneskelige evne til at meddele sig til andre ved hjælp af symboler som fx det talte eller nedskrevne ord. Kigger man nærmere på, hvad der med rimelighed lader sig betegne som kommunikation, så viser det sig hurtigt at dreje sig om ganske mange aktiviteter, der indbyrdes er meget forskellige: Det er kommunikation at lægge en trøstende hånd på en trist vens skulder, og det er kommunikation at publicere en besked på sin Facebook-profil, hvor måske flere hundrede mennesker, fordelt over hele kloden, kan se den.

Gennem sin lange historie har menneskeheden udviklet et støt voksende antal medier, der på forskellig vis kan bruges til at understøtte kommunikation. Megen af den kommunikation, der er foregået i menneskehedens historie, er dog blevet udfoldet uden brug af andre midler end den menneskelige krop, der i sig selv har en række egenskaber, der tillader kommunikation (som fx stemmebånd, ører og øjne).

Udviklingen af forskellige medieformer har imidlertid haft stor betydning for samfundsudviklingen, blandt andet fordi nye medieteknologier historisk set har tilladt kommunikation at forløbe hurtigere og over større afstande, og fordi mange medier også tillader nye måder at oplagre og genkalde information på. Med opfindelsen af skriftsproget blev lagringen af viden gjort mere uafhængig af den menneskelige krop og hukommelse, hvilket letter transmissionen fra generation til generation betydeligt: Før skriftsproget var vidensoverførsel bundet til at foregå mundtligt og ansigt til ansigt, hvilket vanskeligt kan realiseres i større skala. Brugen af skrift gør det også muligt at genkalde informationer mere præcist end ved mundtligt overleverede vidnesbyrd: I orale kulturer (dvs. kulturer uden skriftsprog) indkodes vigtige informationer ofte i rim og remser for at gøre dem lettere at huske. Den digitale computer har gjort lagring, genkaldelse og udveksling af meddelelser muligt i en skala, som langt overskrider noget, der tidligere er set i menneskehedens historie, og på trods af at vi kun har haft adgang til disse muligheder i en (historisk set) meget kort periode, mærkes konsekvenserne allerede i talrige sammenhænge og på mange forskellige niveauer i samfundet.

Kommunikationsbegrebet

Udforskningen af den menneskelige kommunikation er siden 1940'erne foregået under henvisning til en række kommunikationsmodeller. En af de mest indflydelsesrige blev foreslået af den russiske sprogforsker Roman Jakobson (1896-1982).

Kontekst		
Afsender	Meddelelse	Modtager
	Kode	
	Kanal	

(Gengivet efter Jakobson, R. (1960) *Closing statement*. I: Sebeok, T.A. *Style in language*. Cambridge: MIT Press).

Modellen fremstiller kommunikation som en proces, der fundamentalt forløber gennem udveksling af *meddelelser* mellem *afsender(e)* og *modtager(e)*. Årsagen til, at det overhovedet er nødvendigt at udveksle meddelelser, er, at indre tilstande som fx tanker, følelser eller hensigter ikke umiddelbart er tilgængelige for andre end en selv og derfor må bringes på en anden form, hvis man vil delagtiggøre andre i dem.

Den kode, som indgår i modellen, refererer til de systemer af regler og referencer, der bruges, når meddelelserne konstrueres. I talesproget udgøres koden bl.a. af reglerne for, hvordan bestemte lydbilleder kædes sammen med bestemte forestillinger (fx at ordet "hest" refererer til et dyr med fire ben, man kan ride på), samt de grammatiske og syntaktiske regler for, hvordan ord kan kombineres til sætninger. Koden skal kendes af begge parter: Hvis man forsøger at kommunikere med en person, der taler et sprog, man ikke forstår, så kan man nok modtage vedkommendes meddelelser (man kan høre, at de siger noget), men man kan ikke afkode et forståeligt indhold, da man ikke kender den sproglige kode. Der findes flere andre kodesystemer end det verbalsproglige: Således har kropssproget sine egne koder, ligesom visuelle medier som fx tv også betjener sig af konventionelle former.

For at kommunikationen kan forløbe, må der også være en *kanal* til rådighed. Hvis man taler sammen ansigt til ansigt, er kanalen den atmosfæriske luft, som tillader lydbølgerne at brede sig fra stemmebånd til trommehinde. Hvis man bedriver medieret kommunikation, fx via e-mails, er kanalen den digitale infrastruktur, der tillader såvel transport som fremvisning af meddelelserne, dvs. både netværket, e-mail-systemet og de computerterminaler, som afsender og modtager bruger til at læse og skrive på. Udviklingen af den digitale teknologi har betydet, at antallet af tilgængelige kanaler er vokset betydeligt.

I den sammenhæng er det vigtigt at bemærke, at de konkrete teknologiske egenskaber ved kanalen har en betydning for, hvilke koder det er muligt at bruge: Fastnettelefoner tillader fx kun udveksling af lyd, hvorfor det er de koder, der knytter sig til sproglig kommunikation, der bruges. I systemer til video-telefoni kan de kommunikerende både se og høre hinanden, hvilket gør det muligt at inkludere visuelle koderegistre, som fx kropssprog, i kommunikationen. Mange af de nye medier har ført til skabelsen af specielle kodesystemer, eksempelvis i form af sms-sprog med mange forkortelser, smileys og blandinger af bogstaver og tal i samme ord. Blandt teenagere bruges disse specielle koder undertiden så intensivt, at meddelelserne i praksis er uforståelige for udenforstående.

Endelig peger modellen også på *konteksten* som et centralt element i kommunikationsprocessen. Konteksten henviser til forhold og omstændigheder, der har betydning for kommunikationen, men som ikke er en eksplicit del af meddelelsen. Et udsagn som "Det er godt nok surt!" betyder to meget forskellige ting, afhængigt af om man siger det til en ven, der fortæller, at han er ved at miste sit job, eller om man skal fortælle, hvordan en skefuld citronsaft smager. Konteksten inkluderer også mere generelle sociale normer, som vi sjældent tænker over, men løbende indretter vores kommunikation efter: Vi kommunikerer fx meget forskelligt, alt efter om den, vi taler med, er af det modsatte eller det samme køn som os selv. På samme måde kommunikerer vi meget forskelligt, alt efter om vi taler i en offentlig eller en privat sammenhæng.

Udviklingen af nye medier giver os ikke alene nye kanaler, vi kan kommunikere gennem, men påvirker også, hvilke kodesystemer vi har til rådighed, når vi bruger dem. På samme måde kan nye medier også sætte vores vante opfattelser af kontekstuelle forhold under pres: Hvor skellet mellem offentlig og privat tidligere i hovedsagen fulgte fysiske grænser (fx mellem hjemmet og bylivet), så er det ikke uden videre givet, hvor grænsen går på Facebook. Der er sikkert mange, der har prøvet at fortryde ting, de har skrevet på deres Facebook-wall, eller at blive pinligt berørt af fotos fra private situationer, som pludselig er dukket op på nettet.



Åbenheden
for programmering
er central for
udviklingen
af digitale
medier

Hvad er digitale medier?

En kortfattet definition af medier er, at de er tekniske systemer eller redskaber, der sætter os i stand til at meddele os til hinanden. I denne forstand kan et medie opfattes som en afgrænset kanal, der tillader transmissionen af meddelelser mellem en (eller flere) afsender(e) og en (eller flere) modtager(e). Efter denne definition er den digitale, netværksforbundne computer altså ikke i sig selv et medie, da den ikke kan betragtes som en afgrænset kanal. Den kan derimod ses som et metamedie eller som en infrastruktur, der tillader udviklingen af nye medier, der baserer sig på denne infrastruktur. Digitale medier kan således forstås som medier, der benytter sig af en digital infrastruktur, og som tillader udvekslingen af meddelelser mellem forskellige grupper af afsendere og modtagere. E-mails og sms'er samt onlinefora og sociale netværkstjenester kan i den sammenhæng opfattes som digitale medier, ligesom podcasts, netbaseret tv og meget andet. Udviklingen af den digitale computer op gennem anden halvdel af det 20. århundrede har således muliggjort udviklingen af talrige nye medier, hvoraf mange er blevet taget i anvendelse i såvel erhvervslivet og den offentlige forvaltning som i hverdagslivet. Bredden i anvendelsen betyder imidlertid også, at et samlet billede af de digitale medier og alle deres mange, specialiserede anvendelser ikke kan tegnes fyldestgørende i denne sammenhæng. I det følgende vil de digitale medier derfor blive diskuteret med henblik på deres brug blandt almindelige mennesker inden for rammerne af deres hverdagsliv, hvorimod den professionelle brug af digitale medier i fx administrative og organisatoriske sammenhænge vil blive forbigået.

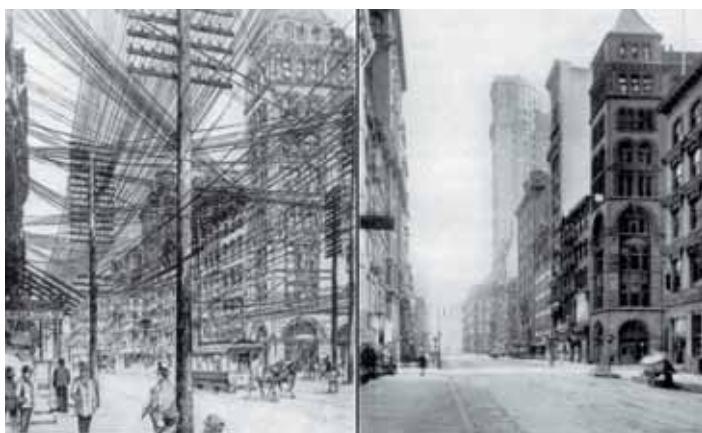
I mediehistorisk perspektiv er de digitale medier specielle, fordi de betjener sig af et fælles, teknologisk grundlag i form af netværksforbundne, digitale computere. Hovedeksemplet er den almindelige computer med forbindelse til internettet, som giver adgang til brugen af bl.a. e-mails, chat og world wide web. Også GSM-baserede mobiltelefoner er i denne sammenhæng at betragte som digitale medier, selvom de første generationer af mobiltelefoner ikke var i stand til at få forbindelse til internettet: Det centrale er, at de baserer sig på digitaliseret information, da denne altid vil kunne bringes over på andre computersystemer og netværk, såfremt fx telefonselskaberne er villige til at åbne for muligheden. De fleste telefoner, der sælges i dag, kan desuden forbindes til internettet, og hvad der i denne sammenhæng er lige så vigtigt, så er en del af de nyere mobiltelefoner åbne for, at der kan udvikles programmer, der er tilpasset dem, og som kan tilgå internettet.

Åbenheden for programmering er central for udviklingen af digitale medier, da det sammen med opkoblingen til et netværk betyder, at nye medier i vid udstrækning kan etableres gennem programmering og ofte med begrænsede investeringer i infrastruktur, ligesom eksisterende digitale medier kan ændres og udbygges uden udskiftning af den samlede infrastruktur. Til forskel herfra har udviklingen af nye analoge medier historisk set været forbundet med udrulningen af separate teknologiske systemer, ofte både til produktion og distribution af kommunikation: Etableringen af tv betød i 1950'erne, at der skulle etableres et system af tv-sendere og produktionsfaciliteter parallelt med det eksisterende radionet. På samme måde krævede etableringen af telefonsystemet i slutningen af 1800-tallet etablering af centraler og udrulning af telefonkabler.

Til forskel herfra betjener både chatsystemer, e-mails, world wide web og onlinespil sig alle sammen af internettet, og når måden at koble sig til internettet på ændrer sig, som fx ved overgangen fra langsomme modemforbindelser til hurtigere, ADSL-baserede bredbåndsforbindelser omkring



Figur 1: Den stærkt kontroversielle musikvideo til Madonna-nummeret 'Like a Prayer' (1989), som oprindeligt kun kunne ses på tv, kan i dag ses døgnet rundt på youtube.com (screenshot fra youtube.com).



Figur 2: Indførelsen af telefonen medførte i første omgang opstilling af talrige telefonmaster. Siden lagde man kablerne under jorden. Billederne viser samme gade i New York ca. 1890 og 1910 før og efter nedgravningen af kablerne.

årtusindskiftet, så får man både hurtigere e-mails, hurtigere downloads af hjemmesider og kortere pingtider til World of Warcraft.

En anden vigtig egenskab ved fleksibiliteten i den digitale infrastruktur er, at den både tillader dannelsen af historisk set nye medieformer (som fx chat) og genskabelsen af allerede eksisterende medieformer, ofte i en næsten identisk form. Eksempelvis kan både tv og radio sendes og modtages via internettet, og indholdet af en musik-cd kan rippes og distribueres via et fildelingsnetværk i en udmærket kvalitet, omend ikke altid lovligt. Det bliver på den måde muligt at flytte indhold fra ét medie til et andet og fra ét distributionskredsløb til et andet, hvilket på den ene side åbner for kulturel og social innovation, men på den anden side skaber problemer med fx ophavsret og betaling for indhold.

Udviklingen af den digitale teknologi har således banet vejen for udviklingen af nye medieformer og for genskabelsen af eksisterende, analoge medier på en digital platform.

Et landkort over digitale medier

Den hastige tilvækst i antallet af nye medier og den meget forskellige udbredelse, de indtil videre har opnået, gør det meget vanskeligt at få et overblik over det samlede felt af digitale medier. En måde at ordne de digitale medier i forhold til hinanden på og i forhold til de eksisterende medier er at opdele dem efter, hvilken form for kommunikation de understøtter.

I den nedenstående model er de digitale medier placeret efter, hvilken kommunikationsform de i hovedsagen benyttes til:

Figur 3: Modellen er udviklet i fællesskab med prof. Klaus Bruhn Jensen.

Seks prototypiske kommunikationsformer		
	Asynkron	Synkron
En til en	Sms'er, e-mails, papirbreve	Chat, telefonsamtale (mobil og fastnet), samtale ansigt til ansigt
En til mange	Hjemmesider, podcasts, aviser, bøger	Tv, radio, online-tv/radio-kanaler
Mange til mange	Sociale netværkssider (Facebook), blogs, fildeling, online-multiplayerspil, opslagstavler	Online-chatrum, forsamlingshusdiskussion, stadion-speak

I modellen skelnes der på den ene side mellem mediernes typiske afsender- og modtagerforhold og på den anden side mellem kommunikationens tidslige forløb. Begge dimensioner er centrale for en given kommunikations konkrete forløb og udformning.

Sondringen mellem synkron og asynkron kommunikation refererer til kommunikationens udstrækning i tid, altså om kommunikationen forløber i realtid eller forskudt. Synkron kommunikation er præget af, at man er nødt til at koncentrere sig om den igangværende kommunikation, hvis man vil have det fulde udbytte af den. Hvis man forlader en animeret diskussion under et middagsselskab i et par minutter, risikerer man at være hægtet af forløbet, når man kommer tilbage, ligesom det er tilfældet, hvis man forlader tv-stuen under en liveudsendelse.

Til sammenligning er asynkron kommunikation mindre flygtig, og man har mulighed for at bladere et par sider tilbage i en bog, hvis man har mistet koncentrationen under læsningen, ligesom man kan gennemlæse tidligere meddelelser i en sms-korrespondance, hvis man har glemt, hvordan diskussionen startede. Sondringen mellem synkron og asynkron indebærer ikke, at den ene type er bedre end den anden, men at de er velegnede til forskellige ting, og at de to typer kommunikation realiseres meget forskelligt. I forhold til kommunikationsmodellen fra faktaboksen kan tidsdimensionen ses som en meget vigtig kontekstuel parameter, da den er afgørende for, om de involverede oplever, at de er del af en fælles situation eller ej.

Man bør være opmærksom på, at ikke alle medier kan klassificeres entydigt med hensyn til denne dimension i kommunikation: Eksempelvis er transmissionstiden for e-mails ofte så kort, at det er muligt at bruge systemet til kommunikation, der kommer meget tæt på det, der kendes fra synkron chat, hvor man skriver korte meddelelser, der er tilgængelige for modtageren næsten omgående efter afsendelsen. Omvendt kan chatsystemer også bruges til asynkron kommunikation. Når de to medier er placeret forskelligt i modellen ovenfor, er det, fordi undersøgelser har vist, at de helt overvejende bruges til hhv. asynkron og synkron kommunikation.

Sondringen mellem en til en, en til mange og mange til mange peger på, at kommunikationsforløb også præges af antallet af deltagere og deres indbyrdes konfiguration som afsendere og modtagere. Der er betragtelig forskel på, om man deltager i en diskussion med sine venner på Facebook, om man går ind på Danmarks Radios website, eller om man fører en e-mail-korrespondance med en ven. Mulighederne for at komme til orde varierer betydeligt, og ens muligheder for at få andre i tale og præge kommunikationsforløbet er meget forskellige i de tre tilfælde.

Intermedialitet

Foruden opdelingen i forskellige kommunikationsformer og den afledte klassifikation af medier, så kan modellen som helhed også give anledning til en yderligere iagttagelse, nemlig at de fleste danskere i dag har frit valg mellem samtlige modellens digitale medier. Internettet har ikke alene givet os adgang til en formidabel vidensressource, med store leksika, internationale aviser og tv- og radiokanaler, men også til interaktion med andre mennesker på kryds og tværs af kloden, i det omfang vi har tid og lyst til at engagere os med dem. De mange parallelle valgmuligheder fordrer også, at vi rent faktisk træffer en række til- og fravalg: De enkelte medier er ikke alene om at facilitere kommunikation, men indgår i en samlet palet af valgmuligheder – en egenskab, der betegnes som intermedialitet. Skabelsen af den digitale infrastruktur har således ikke resulteret i udviklingen af en række medier, der målrettet har ramt eksisterende, opfyldte behov, men har i stedet ført til en situation, hvor de digitale medier eksisterer som parallelle muligheder for at konfigurere kommunikation. Brugere af de digitale medier er som regel udmærket klar over de eksisterende valgmuligheder og etablerer brugsmønstre, der trækker på et begrænset udvalg af de digitale medier. Dertil kommer, at de fleste bruger medier som led i en kommunikationsproces, der har et andet formål end blot selve mediebrugen – sms'er sendes for at meddele sig til andre, og vi lytter til podcasts for at få oplevelser eller lære noget nyt. Anvendelsen af de digitale medier sker således i et samspil mellem brugere, der har en række specifikke formål på den ene side og en række forskellige valgmuligheder på den anden side. I de følgende afsnit skal vi se på en række af de mest markante diskussioner om de digitale mediers muligheder på den ene side og deres faktiske anvendelse på den anden.

En til mange-kommunikation

Opdelingen i kommunikationens afsender- og modtagerforhold peger på, at de nye medier kan bruges til at formidle god, gammeldags massekommunikation: Der er i princippet ikke længere noget, der forhindrer almindelige mennesker i at starte deres egen tv-station. Søger man lidt på internettet, kan man hurtigt finde eksempler på, at der faktisk er nogle, der gør det, fx ved at sætte webcams op i deres dagligstue og lade dem transmittere i døgndrift til alle, der måtte ønske (og



De digitale medier eksisterer som parallelle muligheder

Figur 4: Nettet bruges fortsat i vid udstrækning til formidling af massekommunikation. Her ses Jon Stewarts The Daily Show på DR2 via internettet.



orke) at kigge med, eller ved at producere videopodcasts. Til sammenligning har etableringen af de "gamle" massemedier såsom fx tv eller aviser været forbundet med omfattende anlægsomkostninger og for tv og radios vedkommende også med en høj grad af statslig styring. Udviklingen af de digitale medier er således blevet set som en udfordring af de etablerede massemedier, fordi internettet teknisk set gør det let at etablere parallelle distributionssystemer, der principielt gør det muligt for enhver at konkurrere med de etablerede medier. I praksis har denne konkurrence imidlertid vist sig at være af begrænset omfang, og skønt der er skabt alternative medier online, fx i form af blogs med store læsertal eller rene online-tv-kanaler, så spiller disse medier fortsat en begrænset rolle i de fleste menneskers daglige forbrug af eksempelvis nyheder. Tværtimod har de digitale medier bidraget til en øget anvendelse af en til mange-kommunikation – afsendt fra de etablerede medieinstitutioner: Hjemmesiderne fra Danmarks Radio, TV 2, Politiken og Jyllands-Posten hører således til i toppen af danskernes foretrukne hjemmesider. En væsentlig årsag til denne tilsyneladende konservatisme er, at de etablerede medier leverer indhold af en høj kvalitet i formater, der er letgenkendelige, og at de etablerede medier nyder en høj grad af troværdighed i befolkningen – noget, som mange ikke umiddelbart tilskriver meget andet indhold online.

Mange til mange-kommunikation

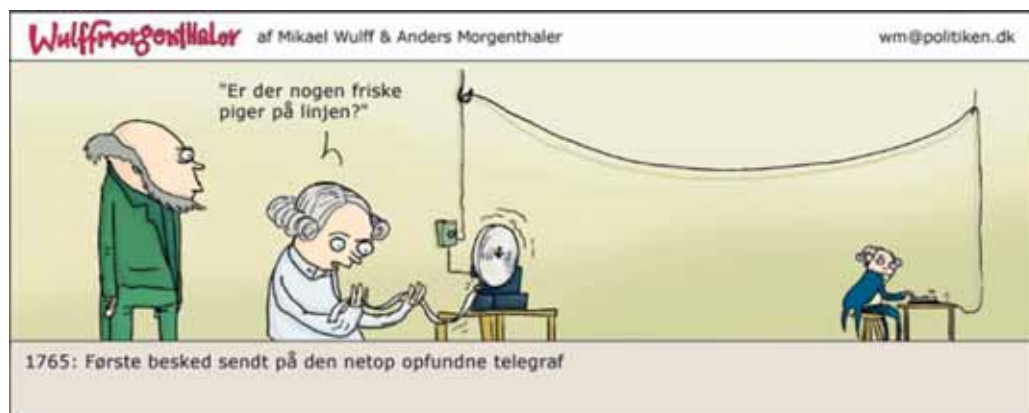
Enkelte af de allermest benyttede sites på internettet tilbyder ikke massekommunikation, men derimod mange til mange-kommunikation, hvilket er bemærkelsesværdigt, da denne kommunikationsform ikke har nogen klare forløbere i mediehistorien. Det betyder (jf. faktaboksen), at koderne for udfoldelsen af denne kommunikationsform har måttet skabes fra bar bund. Problemerne med at klassificere kommunikationen på eksempelvis Facebook som enten offentlig eller privat er allerede omtalt og er typiske for et nyt medie, der er ved at finde en stabil kulturel form. Mulighederne for misforståelser er mange, så længe der ikke eksisterer kontekstuelle normer og koder for, hvordan mediet bruges, og hvordan kommunikationen i det skal opfattes.

”

*Mulighederne
for misfor-
ståelser er
mange*

Til sammenligning er introduktionen af fx e-mails og mobiltelefoni utvivlsomt blevet lettet betydeligt af, at mange brugere havde et repertoire af koder fra meget lignende medier at trække på. Dette er imidlertid ikke tilfældet med mange til mange-kommunikationen, hvor først onlinefora og siden også de sociale netværkstjenester har tilbudt denne form for kommunikation.

Lige siden internettets folkelige gennembrud i starten af 1990'erne har gruppekommunikationen tiltrukket sig meget stor opmærksomhed, også selvom det først er inden for de sidste cirka fem år, at den for alvor har opnået stor udbredelse, bl.a. gennem sociale netværk som Facebook. Her har omkring halvdelen af den danske befolkning i dag oprettet en profil, og Facebook er samtidig det enkeltsite, danskerne tilbringer mest tid på: I skrivende stund lægger sitet beslag på 18% af al den tid, danskerne bruger til private aktiviteter online. Den tidlige interesse for gruppekommunikation online tog i høj grad udgangspunkt i, at denne kommunikationsform blev anset for at have et demokratisk potentiale, og kun i ringe grad i analyser af den faktiske anvendelse af mange til mange-medierne. Udgangspunktet var i stedet de potentialer, man så i mulighederne for udviklingen af en online-debatkultur, der ikke var underlagt de samme restriktioner og krav som dannelsen af den offentlige mening i massemedierne. Her foregår dialogen for en stor dels vedkommende mellem professionelle meningsdannere og er underlagt en udtalt grad af redaktionel kontrol. I modsætning hertil blev onlinefællesskaberne set som en arena, hvor hvem som helst kunne komme til orde, og hvor den demokratiske debat kunne finde en mere komplet og uhildet form.



Figur 5: Brugen af nye medier følger ofte brugernes umiddelbare behov og interesser tættere, end man måske bryder sig om at tænke på. Bemærk, at telegrafens opfindelse på tegningen dateres 72 år for tidligt.

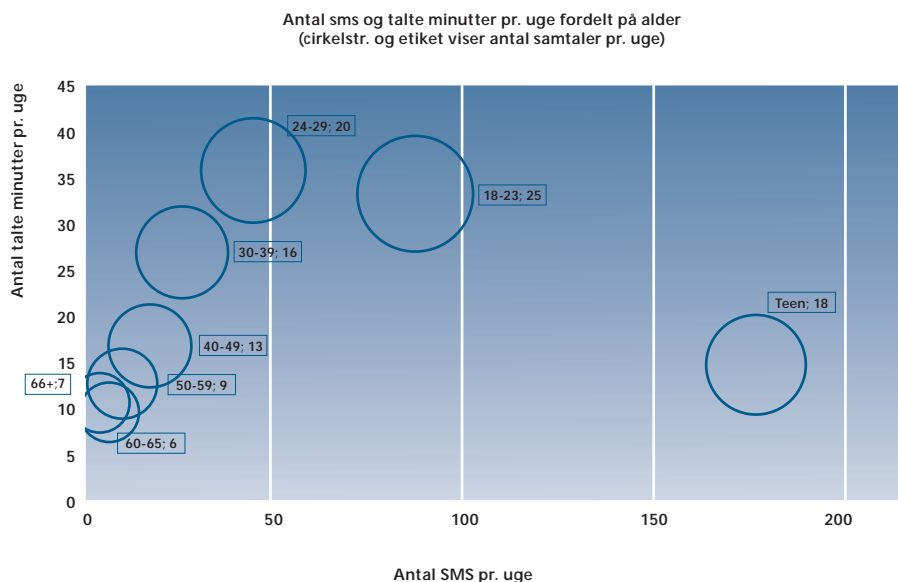
Kvantitativt set har den faktiske brug af mulighederne for gruppekommunikation online imidlertid i høj grad været domineret af andre hensyn end de rent demokratiske – nettet er i vid udstrækning blevet brugt til kommunikation om emner af en mere umiddelbar og ordinær karakter, ikke mindst til vedligeholdelsen af eksisterende sociale relationer og eventuelt til indgåelsen af nye. Dermed er det ikke sagt, at internettet ikke har et demokratisk potentiale, hvilket i denne sammenhæng kommer til udtryk på to måder. Dels har mange organisationer, som fx politiske græsrodsbevægelser, fået nye muligheder for både intern og udadrettet kommunikation, hvilket kan have en demokratisk funktion af den type, man oprindeligt var interesseret i. Dels kan man argumentere for, at mulighederne for at interagere på sociale netværkssites i sig selv har

en samfundsstyrkende karakter, uanset om interaktionen drejer sig om politiske emner eller ej. Sociale relationer kan i visse tilfælde vedligeholdes og styrkes blot ved at blive brugt, og de sociale netværkssider kan for eksempel gøre det muligt at bevare en sporadisk kommunikation med personer, man ikke længere omgås, og som man ellers helt ville miste kontakten til.

De store forventninger til internettets demokratiske potentialer understreger en central pointe, nemlig at nye medier først opnår deres sociale betydning, når de tages i brug i konkrete sociale sammenhænge, samt at de sjældent tages i brug på måder, der fuldstændig bryder med etablerede mønstre i disse sammenhænge. Det er langt mere almindeligt at se, at nye medier bruges med udgangspunkt i allerede etablerede kommunikationsmønstre, som de kan være med til at udvikle og på den måde også forandre. På den led kommer den faktiske indlejring af de digitale medier i kulturen til at forløbe i nært samspil med eksisterende sociale og kulturelle mønstre og eksistensen af andre mediemæssige valgmuligheder. Det følgende afsnit viser, hvordan også brugen af mobiltelefonens to medier, sms og telefoni, indgår i et nært samspil med brugernes livssituation.

En til en-kommunikation

Mobiltelefonen er, som teknologi betragtet, i praksis udbredt til hele befolkningen, da der findes mindst et eksemplar i 98 % af alle husstande, hvilket betyder, at det er lige så sjældent at finde et hjem uden en mobiltelefon som at finde et uden tv. Blandt de under 60-årige bruges både sms og mobiltelefoni desuden af stort set alle, mens der er ca. 30 % af befolkningen over 60 år, der ikke bruger sms. Ikke desto mindre er der meget store forskelle på anvendelsesgraden af de to medier, der varierer på måder, der viser sig at være meget karakteristiske for brugernes alder og livssituation.



Figur 6: Brug af sms og mobiltelefoni ift. alder.

Figuren viser, at teenagere har en meget klar præference for sms-kommunikation; faktisk taler teenagere så lidt i deres mobiltelefoner, at man skal op i gruppen af de over 40-årige, før man finder nogle, der taler lige så lidt som teenagerne. Figuren viser også, at personer mellem 20 og 30 år har en klar præference for at tale i deres mobiltelefon. En typisk 24-årig taler næsten tre gange så meget i sin telefon som en teenager, der til gengæld sms'er næsten fire gange så meget.

Forskning viser, at det mønster, der tegner sig i figuren, er dynamisk, sådan at brugen af de to medier ændrer sig, efterhånden som vi bliver ældre. Teenagerne lægger gradvis sms-brugen bag sig og øger i stedet brugen af mobiltelefoni, efterhånden som de kommer op i 20'erne, mens folk på den anden side af 30 gradvis sænker deres forbrug af begge medier, efterhånden som de bliver ældre. Der er således en tydelig forskel på, hvilket af de to medier der foretrækkes af personer, der befinder sig forskellige steder i livet.

Teenagere bruger hyppigt sms-mediet til både en til en- og mange til mange-kommunikation: Sms'er kan sendes til et valgfrit antal modtagere, og mens det blandt folk over 18 næsten udelukkende bruges som et en til en-medie, så står gruppe-sms'er for cirka en tredjedel af teenagerens forbrug. Denne brug af mediet illustrerer et karakteristisk træk ved samværsformerne i denne aldersklasse, hvor kommunikationen med en større gruppe af venner, klassekammerater og bekendte spiller en central rolle. Sms-mediet har udviklet sig til at være et centralt (og omfattende) element i de fleste teenageres liv, hvor de fleste typisk sender omkring 25 om dagen og desuden modtager lige så mange, hvilket betyder, at de bruger mediet omkring 50 gange pr. dag.

På trods af denne intensive brug, som for manges vedkommende udgør en fast, daglig rutine gennem flere år, så ændrer mønstret sig drastisk, når folk kommer op i 20'erne, hvor den intense sms-brug viger til fordel for en opprioritering af telefoni. Brugsmønstret fra teenageårene er altså ikke fastere, end at de fleste kan lægge det om, når livsomstændighederne ændrer sig: Den periode, hvor sms-brugen nedtrappes, er samtidig det tidspunkt, hvor de fleste flytter hjemmefra og etablerer deres egen husstand, enten alene, med en partner eller sammen med andre. Samtidig er der også mange, der udskifter store dele af deres sociale omgangskreds i netop disse år i forbindelse med flytning til andre byer og lign., og den erstattes åbenbart ikke uden videre med en ny af tilsvarende størrelse og med samme intensive kommunikationsmønster.

Til forskel fra teenagerne afspejler brugen blandt de unge i 20'erne et kommunikationsmønster, hvor den medierede kommunikation gradvis koncentrerer sig omkring en støt mindre kreds af personer, der typisk udgøres af en partner og nogle venner og familiemedlemmer, som man især holder kontakt med gennem telefonsamtaler. Omlægningen til et hverdagsliv, der i højere grad er opdelt mellem fritid og arbejde, og som byder på flere tidskrævende forpligtelser og ansvarstyper, levner mindre plads til sms-kommunikationens talrige afbrydelser. Ved at lægge kommunikationen om til telefoni bliver det muligt for folk i 20'erne at bevare en relativt høj intensitet i udvalgte relationer og samtidig gøre plads til nogle stræk af relativt ubrudt tid hen over dagen. Hertil kommer naturligvis kommunikation via sociale netværkstjenester, som også spiller en rolle i disse aldersgrupper, men i forhold til de her angivne tal er denne kommunikation ikke intensiv nok til at rykke billedet afgørende.

”

Teenagere har en meget klar præference for sms-kommunikation

Brugen af de to medier viser imidlertid meget konkret, hvordan valget mellem forskellige digitale medier tydeligt er styret af kontekstuelle faktorer, og hvordan et medie, der er en naturlig del af hverdagen på ét tidspunkt i livet, få år efter kan være erstattet af et andet, som tillader et andet miks af kommunikation og kontekst.

Opsamling

Udviklingen af den digitale infrastruktur har muliggjort skabelsen af nye digitale medier, som tilsammen udgør en række parallelle valgmuligheder, der i dag står til rådighed for størsteparten af befolkningen. Brugen af de digitale medier er på den ene side præget af valgfrihed og på den anden side af en række etablerede kulturelle og sociale mønstre, som præger anvendelsen af disse nye medier på flere centrale områder. Den digitale medierevolution tegner sig i dag snarere som en langstrakt udvikling, der finder sted i et intimt samspil mellem nye medier og et gammelt samfund. ❖

Læs mere

Finnemann, N. O. (2005). *Internettet i mediehistorisk perspektiv*. Frederiksberg: Samfundslitteratur.
Helles, R. (2009). *Personlige medier i hverdagslivet*. Københavns Universitet, København.
Jensen, K. B. (2008). *Medier og Samfund. En introduktion*. Frederiksberg: Samfundslitteratur.

Rasmus Helles



Rasmus Helles er adjunkt i medievidenskab på Københavns Universitet, Institut for Medier, Erkendelse og Formidling (MEF). Han blev i 2005 cand.mag. i medievidenskab og ph.d. i 2009 med afhandlingen "Personlige medier i hverdagen".

Rasmus' primære forskningsområde er de nye medier og deres brugere, og han er desuden tilknyttet EU-forskningsprojektet Mediadem,

der kortlægger den mediepolitiske regulering af både nye og traditionelle medier i 14 EU-lande. Han deltager i udviklingen af den nye bacheloruddannelse i kommunikation og it, der kombinerer humanistisk og samfundsvidenskabelig kommunikationsforskning med datalogi. På uddannelsen, der skabes i samarbejde mellem DIKU og MEF, underviser han bl.a. i grundlæggende kommunikationsteori og empiriske undersøgelsesmetoder. Af udvalgte publikationer kan nævnes "Medium Theory" (2009) og "Hverdagslivets nye medier" (2010).

PROSA

Forbundet af It-professionelle



Kubulus er KU's alumneforening for nuværende og tidligere studerende. Kubulus har til formål at skabe sociale og faglige netværk for sine alumner. Læs mere på kubulus.ku.dk.




The Open Platform Company



How the future looks is up to **YOU!**

Milestone Systems is the global industry leader providing true open platform IP video management software and a future-proof security solution. Our XProtect™ product delivers powerful surveillance that is easy to manage, and proven in over 50,000 customer installations around the world. XProtect provides best-of-breed solutions to 'video enable' organizations – reducing costs, optimizing processes, protecting people and assets. Headquartered in Denmark and with ten offices around the world, Milestone and its open platform is poised for unlimited future expansions. www.milestonesys.com

Getting satisfaction from what you do. Just another day at the office for a high performer.

Choose Accenture for a technology career where the variety of opportunities and challenges allows you to make a difference every day. A place where you can develop deep technical skills using the latest software and tools, and work with the smartest people worldwide to do what hasn't been done before. The only place where you can work with our global clients to create and deliver innovative solutions that help them achieve high performance.

If this is your idea of a typical working day, then Accenture is the place to be.

Visit accenturejob.dk
• Consulting • Technology • Outsourcing



accenture
High performance. Delivered.

TILLYKKE MED DE 101000!

```
C:\>Binær dramatik eller  
kedelige decimaler - vi håber,  
at I vil uddanne uundværlige  
kandidater i mindst 40 år endnu...
```

```
C:\>NNIT
```

nnit

Conscience driven. Value adding

Development & IT

Innovative produkter
Spændende arbejdsopgaver

Læs mere

www.danskebank.com/job

Danske Bank

Med **Den digitale revolution – fortællinger fra datalogiens verden** udkommer for første gang i bogform en bred introduktion til aktuel datalogisk forskning. Bogen udgives i anledning af, at Datalogisk Institut, Københavns Universitet (DIKU) blev oprettet i 1970 og dermed runder de 40 år som selvstændigt institut. I bogens artikler gives eksempler på, hvordan datalogien har været med til at revolutionere verden gennem de seneste 40 år.

Bogen giver bl.a. svar på spørgsmål som: Hvorfor bliver computere hurtigere og hurtigere? Hvordan fungerer programmeringssprog? Hvordan kan vi simulere fysiske processer? Kan en computer nogensinde fungere som en menneskelig hjerne? På det mere praktiske plan giver bogen indblik i nyeste principper for menneskers interaktion med computere og fortæller om, hvordan vi sikrer os bedst mod, at uvedkommende får adgang til vores personlige data.

Artiklerne spænder vidt og kan læses uafhængigt af hinanden. Bidragyderne er forskere med tilknytning til instituttet. Formålet er at styrke kendskabet til datalogisk forskning – og mere generelt it-forskning – i det danske samfund. Specifikt er bogen målrettet de gymnasiale uddannelser. Bogen er tilgængelig i pdf-format på DIKUs hjemmeside www.diku.dk